

EpiFi

An In-Home IoT Architecture for Epidemiology Research

Philip Lundrigan, Kyeong T. Min, Neal Patwari, Sneha Kumar Kasera, Kerry Kelly, Jimmy Moore, Miriah Meyer, Scott C. Collingwood, Flory Nkoy, Bryan Stone, and Katherine Sward



EPIFI

- ▶ Epidemiology: study of diseases and it affects populations
- ▶ Goals:
 - ▶ Bring benefits of IoT to medical applications
 - ▶ Allow families, doctors, and researchers to process data in real-time
 - ▶ Designed to be used by other people
 - ▶ Deploy in thousands of homes
- ▶ Part of \$5.5 million NIH grant called Pediatric Research using Integrated Sensor Monitoring Systems (PRISMS)

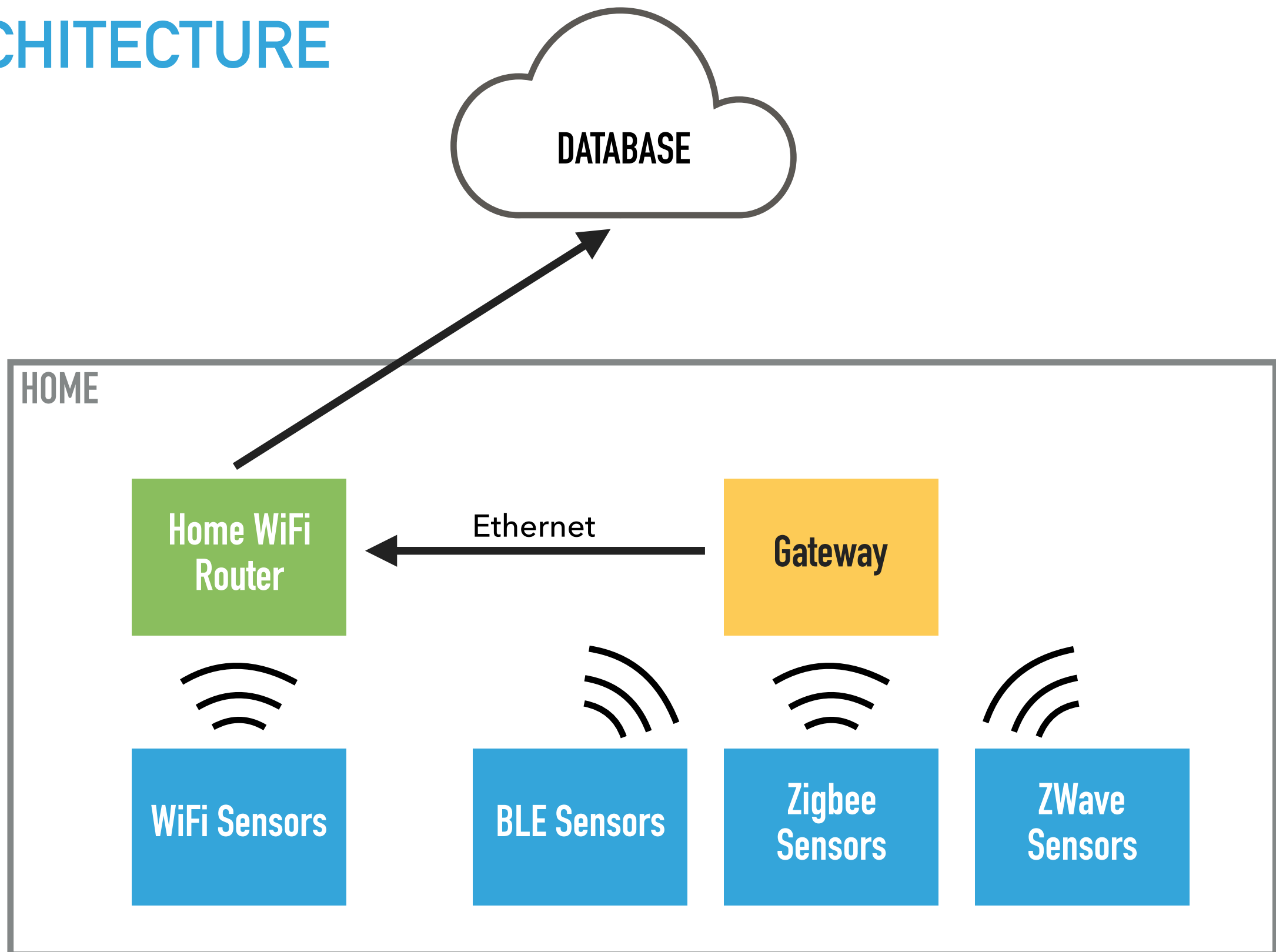
WHY EPIFI?

- ▶ Current IoT solutions are not designed for epidemiology researchers:
 - ▶ Systems not designed for large study based deployments
 - ▶ Cost of management > cost of devices
 - ▶ Must be HIPAA compliant for data transmission and storage



<https://goo.gl/wqKoyz>

ARCHITECTURE



BLE SENSORS

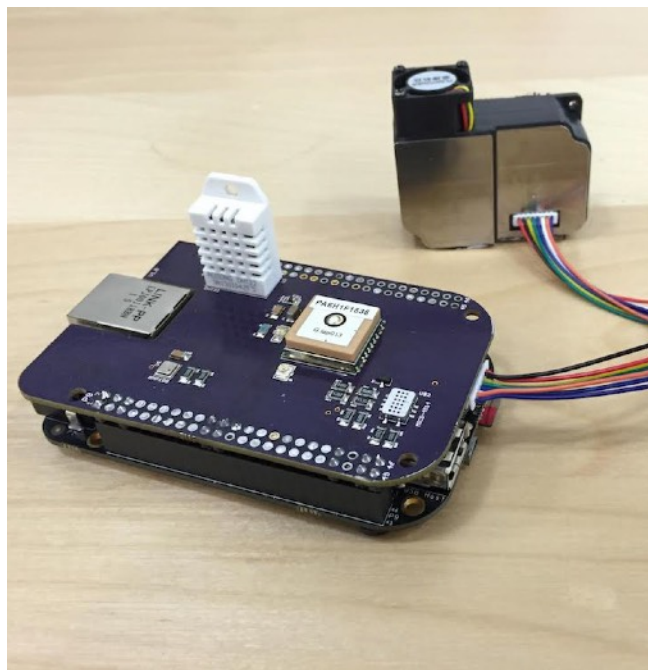


George Washington
University prototype
sensor



Arizona State University
prototype sensor

WIFI SENSORS



AirU



Utah Modified Dylos

FOCUS ON WIFI SENSORS

- ▶ WiFi hardware is readily available and inexpensive
- ▶ WiFi is more widely deployed compared to other wireless protocols
- ▶ A WiFi sensor can integrate with the rest of the home because it uses IP

DEPLOYMENTS

- ▶ Deployed in homes, labs, and an Air Force hangar
- ▶ Used in four studies
 - ▶ A clinical study has been deployed for more than a year with 10 participants
 - ▶ Another study looked at how running furnace fan affects air quality

FOCUS

- ▶ Ease of use
 - ▶ Adding new sensors
 - ▶ Integrating existing sensors
 - ▶ Setting up server infrastructure
- ▶ Reliable data transfer

FOCUS

- ▶ **Ease of use**
 - ▶ Adding new sensors
 - ▶ Integrating existing sensors
 - ▶ Setting up server infrastructure
- ▶ Reliable data transfer

ADDING NEW SENSORS

- ▶ Custom Linux image for sensors
- ▶ Configuration options placed in one file
- ▶ Place sensor specific code in one place and EpiFi takes care of connectivity and reliability transferring the data



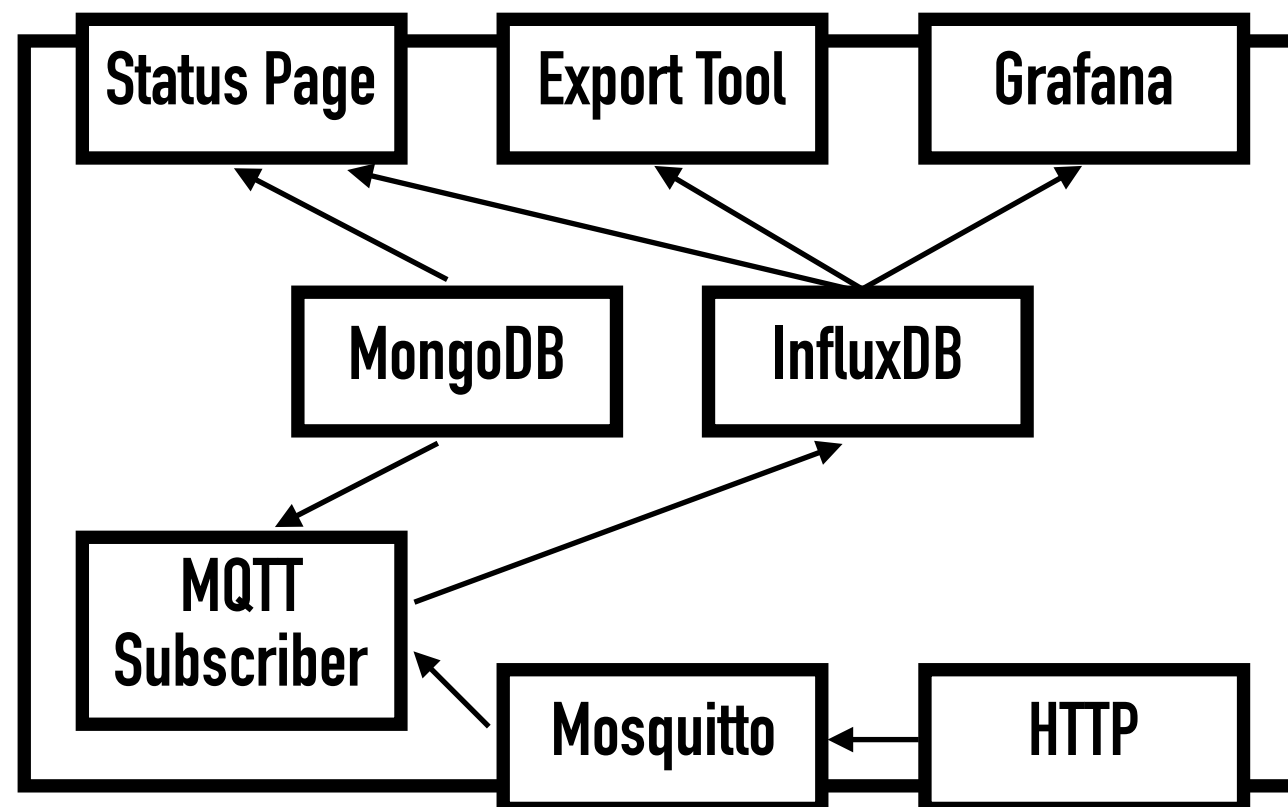
INTEGRATING EXISTING SENSORS

- ▶ Use MQTT or HTTP to upload data to servers
- ▶ Open format

```
{  
    "measurement_name": [value, "unit"],  
    ...  
}
```


SETTING UP SERVER INFRASTRUCTURE

- ▶ Dockerized all server-side components of system
- ▶ Use docker compose to deploy components to a server easily
- ▶ Wrote script to bootstrap databases and set up authentication



FOCUS

- ▶ Ease of use
 - ▶ Adding new sensors
 - ▶ Integrating existing sensors
 - ▶ Setting up server infrastructure
- ▶ **Reliable data transfer**

RELIABLE DATA TRANSFER

- ▶ Epidemiology research needs all historical data
- ▶ First test deployment saw a lot of data loss even though we were using TCP
- ▶ Homes are hazardous environments for wireless sensors
- ▶ Persist data at every opportunity

PROBLEMS ENCOUNTERED

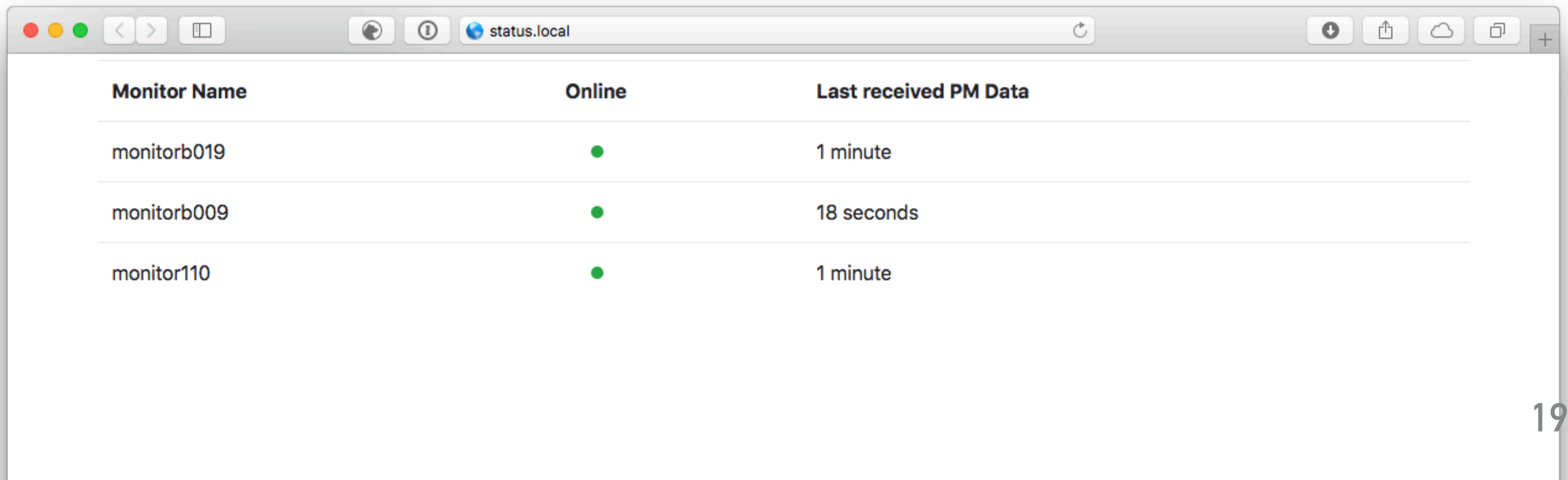
- ▶ Having other people use our system lead to many interesting challenges:
 - ▶ Managing deployments
 - ▶ Bootstrapping WiFi connectivity
 - ▶ Device observability
 - ▶ Data privacy when moving sensors
- Solved**
- Future work**

PROBLEMS ENCOUNTERED

- ▶ Having other people use our system lead to many interesting challenges:
 - ▶ **Managing deployments**
 - ▶ Bootstrapping WiFi connectivity
 - ▶ Device observability
 - ▶ Data privacy when moving sensors
- Solved**
- Future work**

MANAGEMENT

- ▶ Export data for analysis
- ▶ Sensor metadata
- ▶ Monitor status of sensors



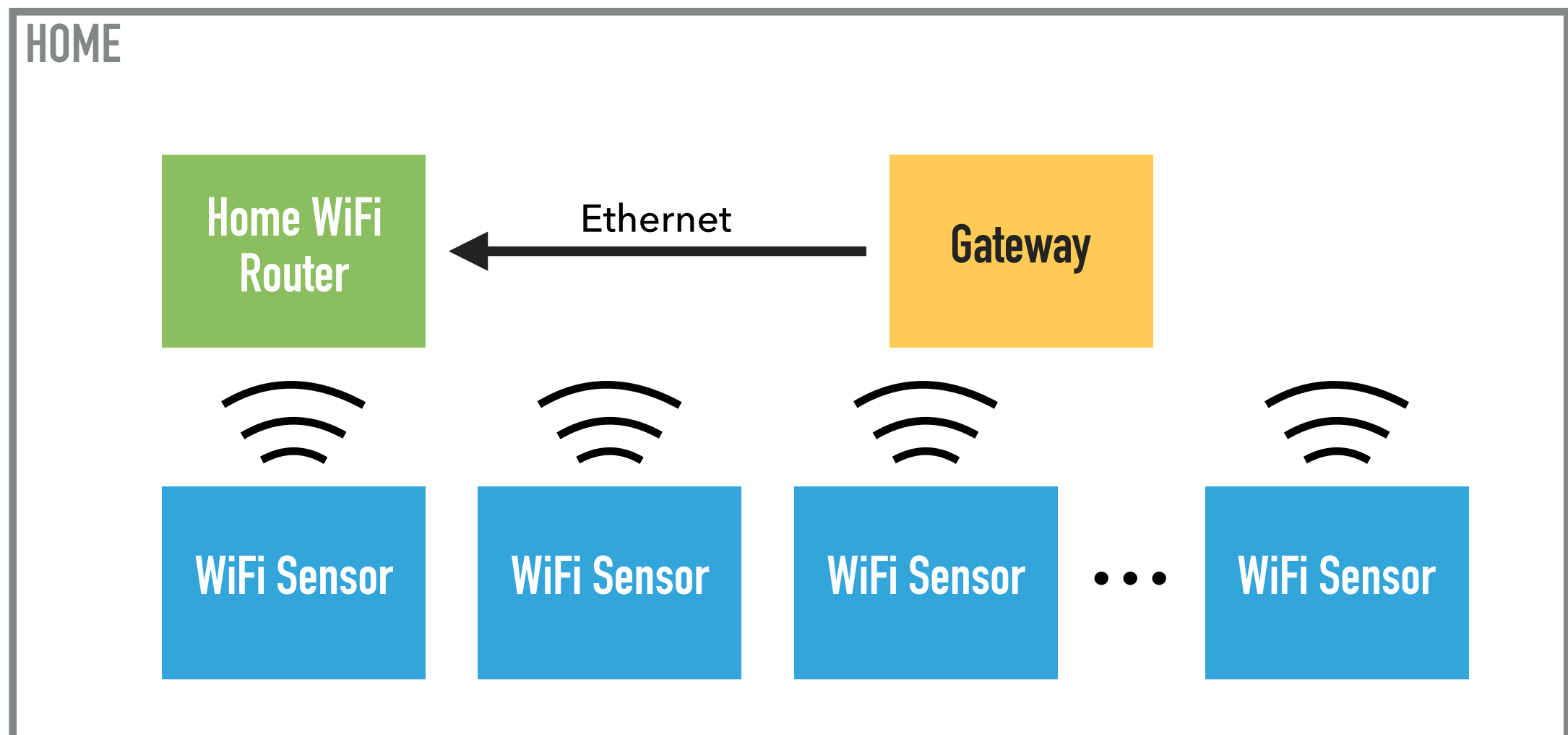
A screenshot of a web browser window displaying the status.local interface. The browser's address bar shows 'status.local'. The page content features a table with three columns: 'Monitor Name', 'Online', and 'Last received PM Data'. The table lists three monitors: 'monitorb019', 'monitorb009', and 'monitor110'. Each monitor is marked as 'Online' with a green dot, and their last received PM data is shown as '1 minute', '18 seconds', and '1 minute' respectively.

Monitor Name	Online	Last received PM Data
monitorb019	●	1 minute
monitorb009	●	18 seconds
monitor110	●	1 minute

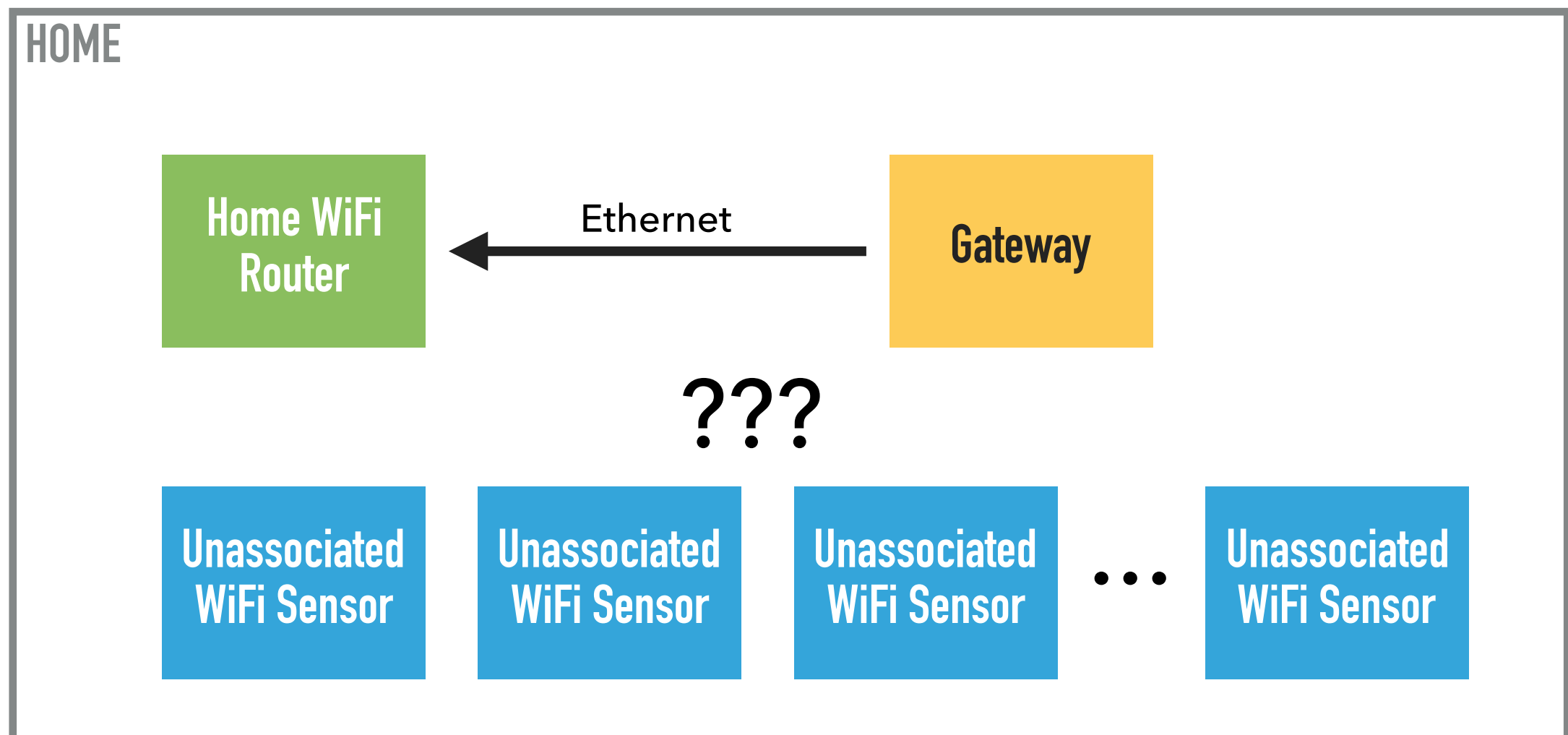
PROBLEMS ENCOUNTERED

- ▶ Having other people use our system lead to many interesting challenges:
 - ▶ Managing deployments
 - ▶ **Bootstrapping WiFi connectivity**
 - ▶ Device observability
 - ▶ Data privacy when moving sensors
- Solved**
- Future work**

SECURE WIFI BOOTSTRAPPING PROBLEM



HOW DO YOU CONNECT A WIFI SENSOR TO A WIFI ROUTER?



HOW DO YOU CONNECT A WIFI SENSOR TO A WIFI ROUTER?

- ▶ Program sensor with home's network and password
- ▶ Bring our own wireless router
- ▶ Open network
- ▶ WPA Enterprise
- ▶ WiFi Protected Setup (WPS)
- ▶ Out-of-band channel
- ▶ SmartConfig
- ▶ WiFi device becomes a temporary AP

HOW DO YOU CONNECT A WIFI SENSOR TO A WIFI ROUTER?

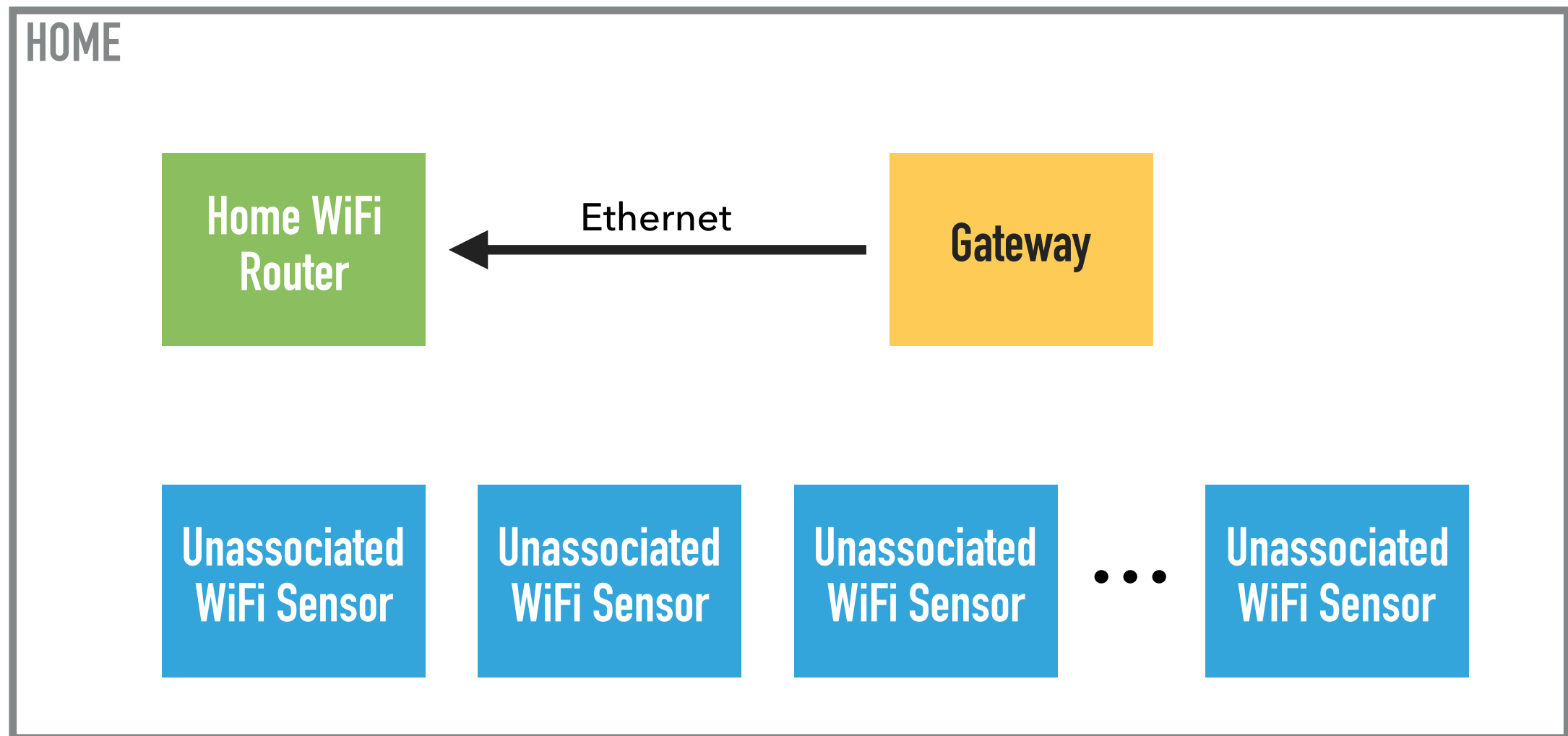
- ▶ Program sensor with home's network and password
- ▶ Bring our own wireless router
- ▶ Open network
- ▶ WPA Enterprise
- ▶ WiFi Protected Setup (WPS)
- ▶ Out-of-band channel
- ▶ SmartConfig
- ▶ **WiFi device becomes a temporary AP**

REQUIREMENTS

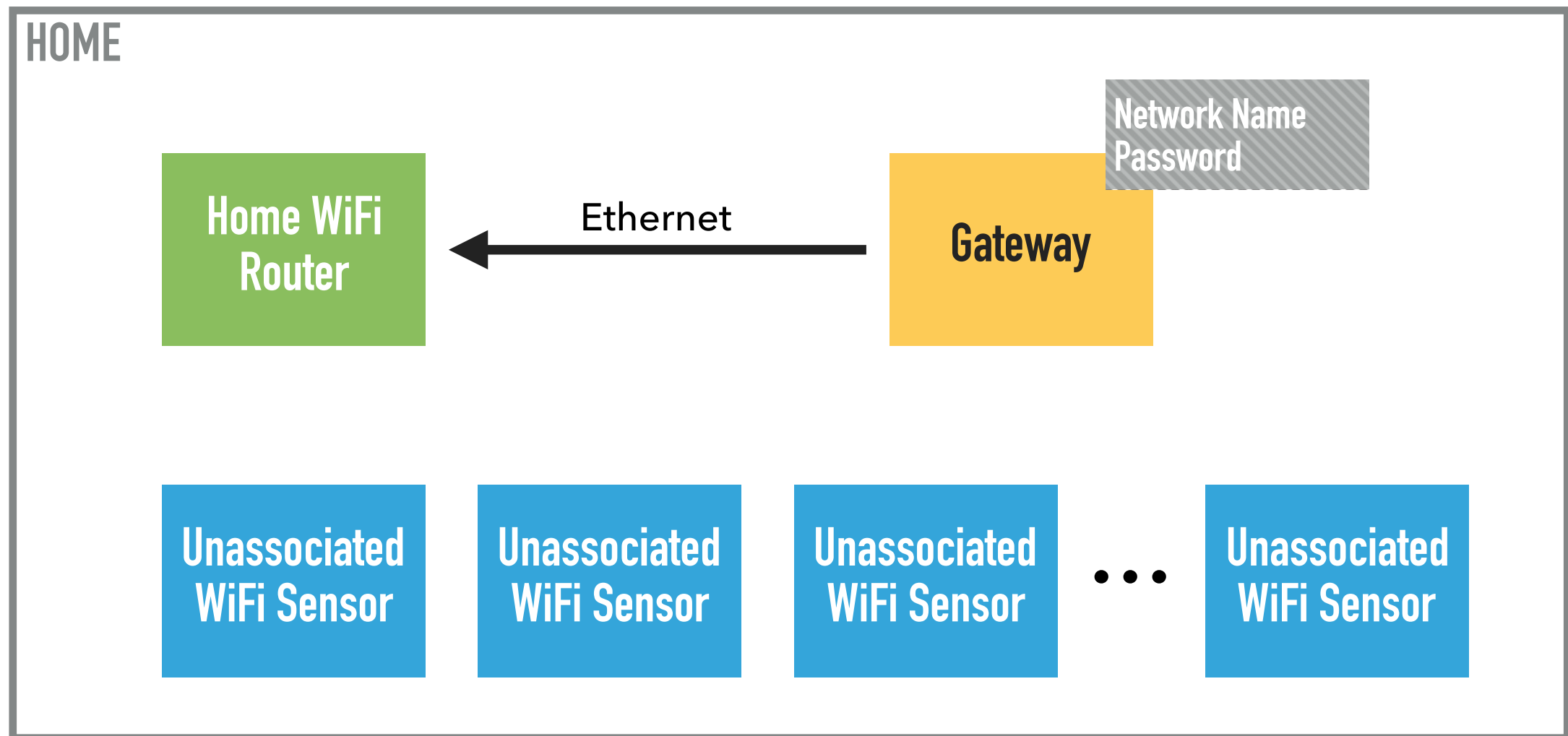
- ▶ Does not require any extra hardware at the sensor
- ▶ Broadly supported by home WiFi routers
- ▶ Time to connect scales up to many devices

**Can we *securely* bootstrap WiFi connectivity
of *many devices* using *just* WiFi?**

SECURE TRANSFER OF AUTHENTICATION PROTOCOL (STRAP) OVERVIEW

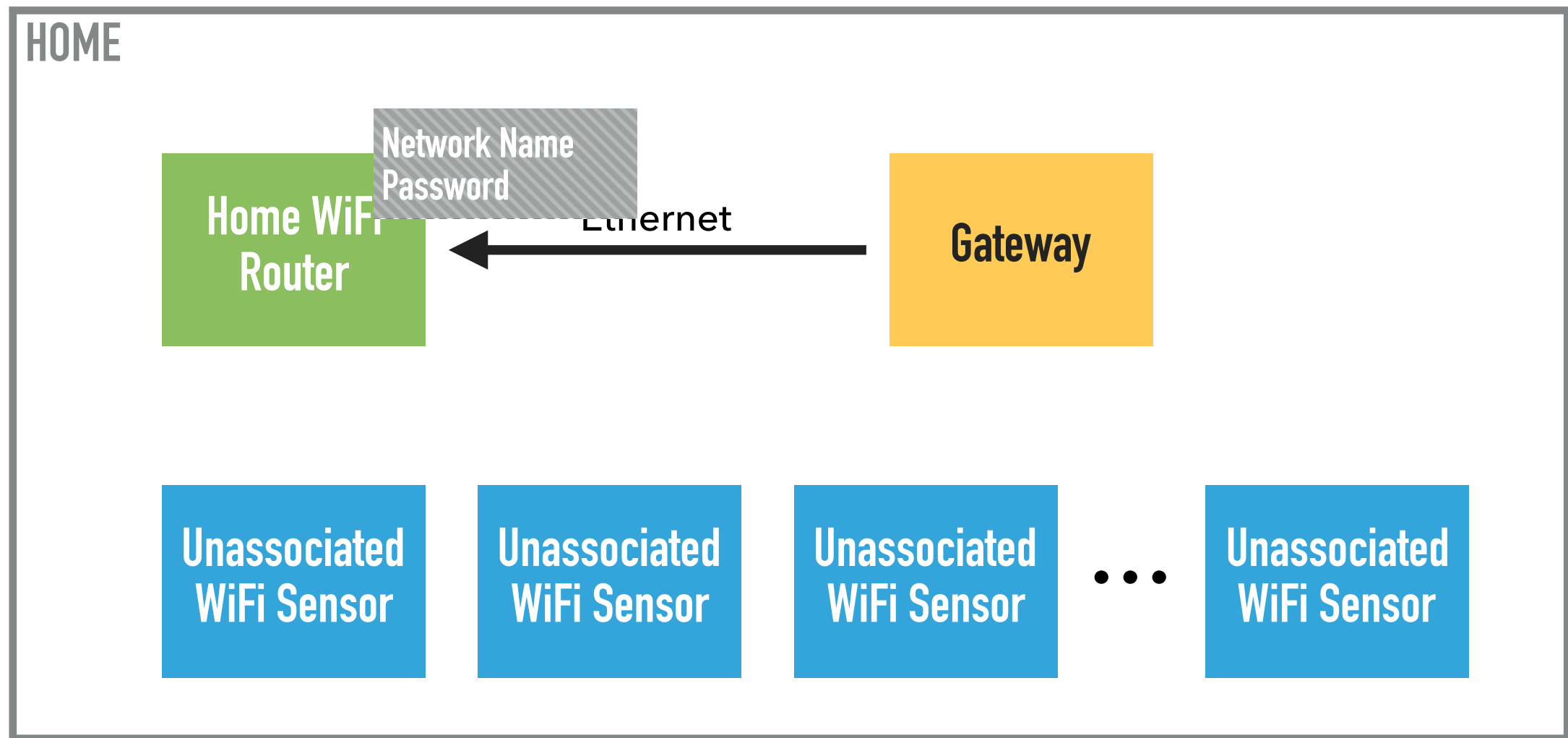


STRAP OVERVIEW



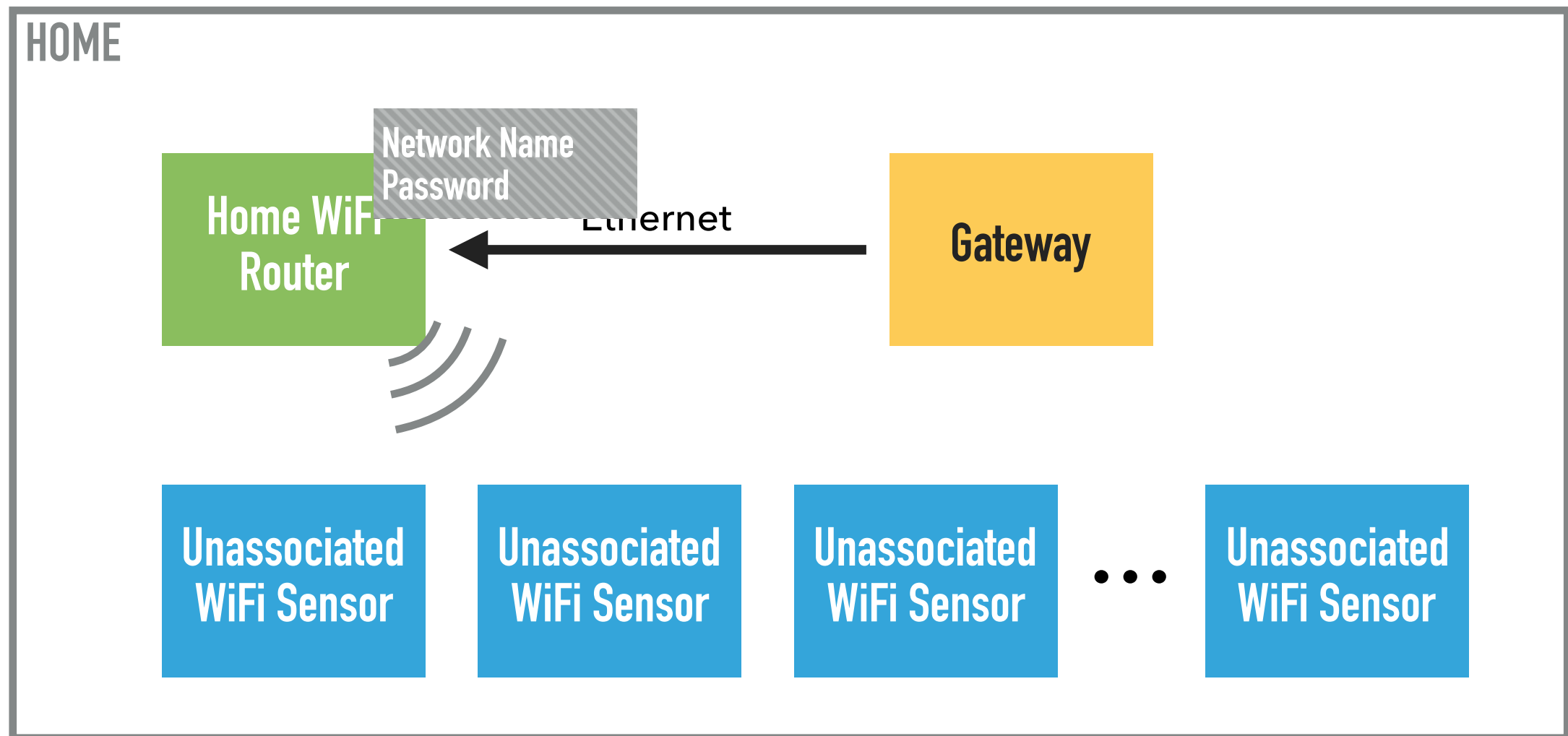
Participant enters network name and password into web interface on gateway

STRAP OVERVIEW



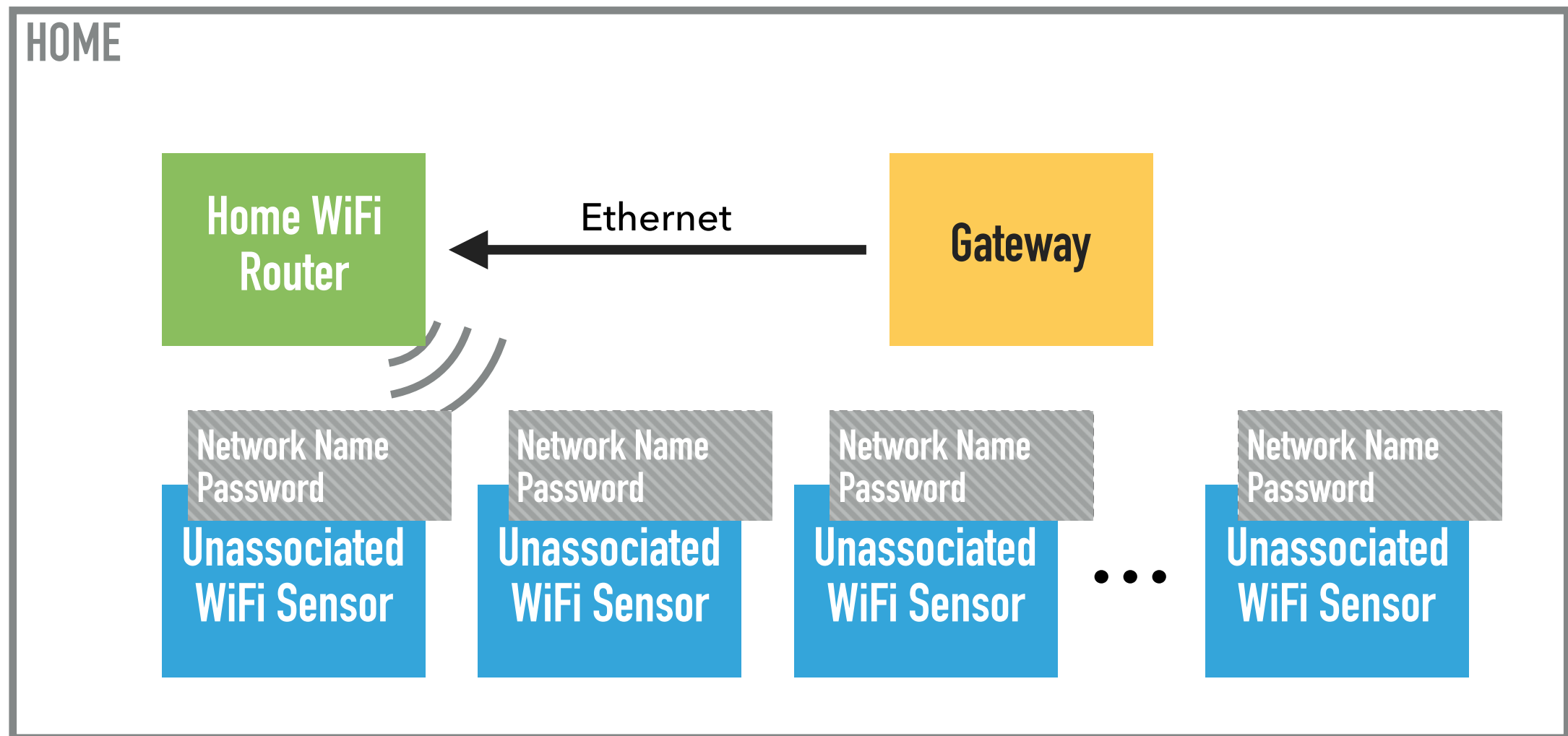
Gateway securely sends network name and password to the home's WiFi router

STRAP OVERVIEW



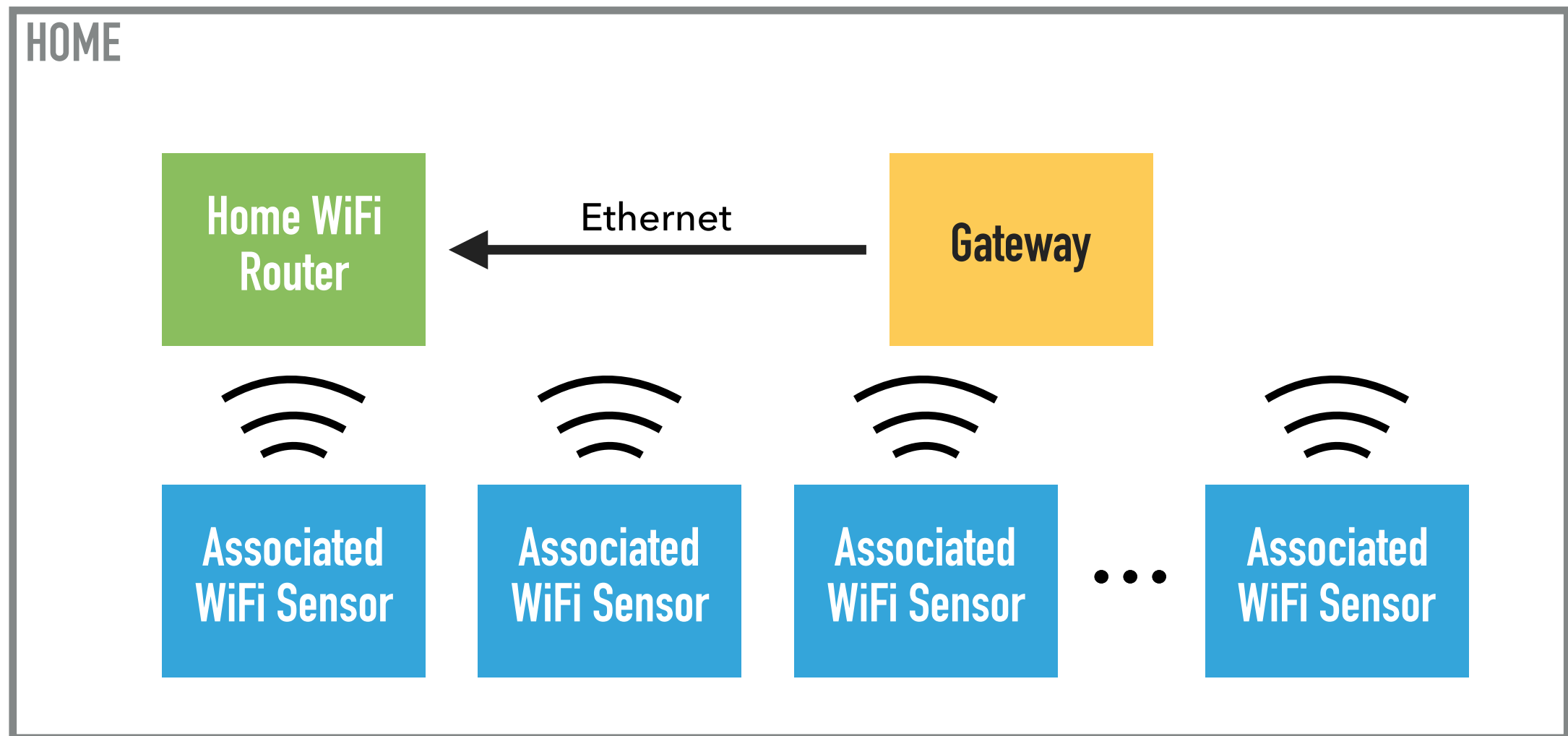
Home WiFi router securely sends network name and password to the *unassociated* sensors

STRAP OVERVIEW



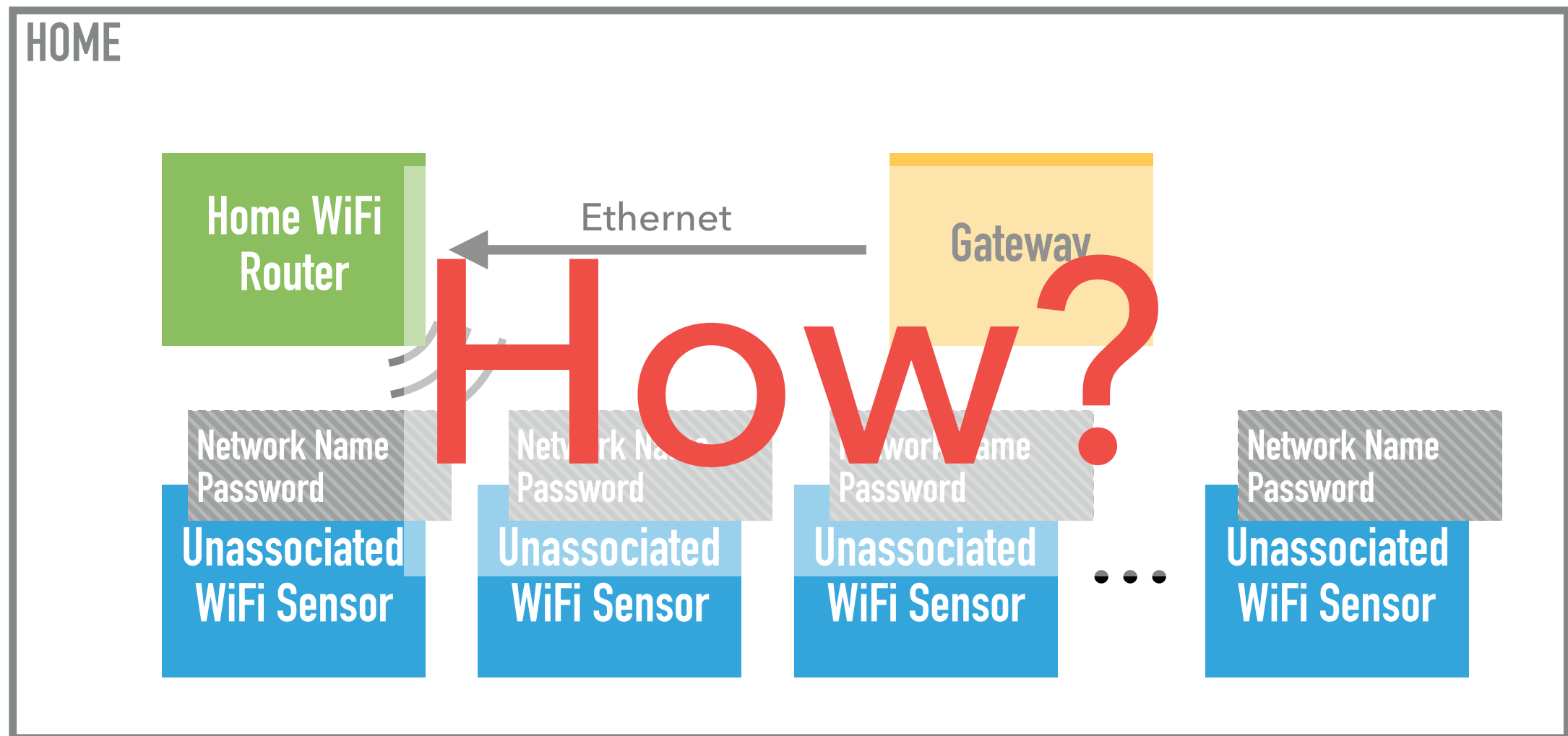
Home WiFi router securely sends network name and password to the *unassociated* sensors

STRAP OVERVIEW



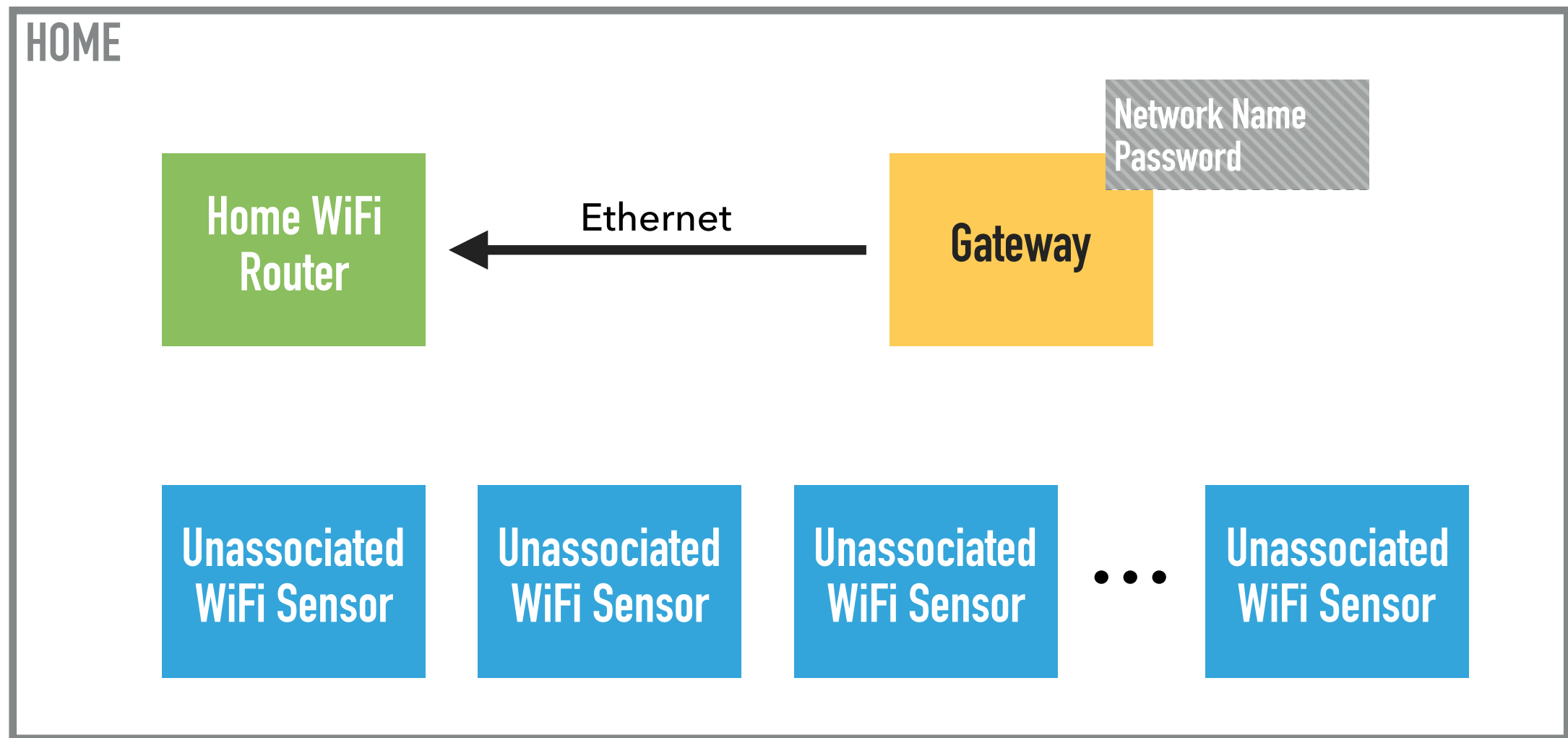
Sensors connect to the network and now can send measurements

STRAP OVERVIEW



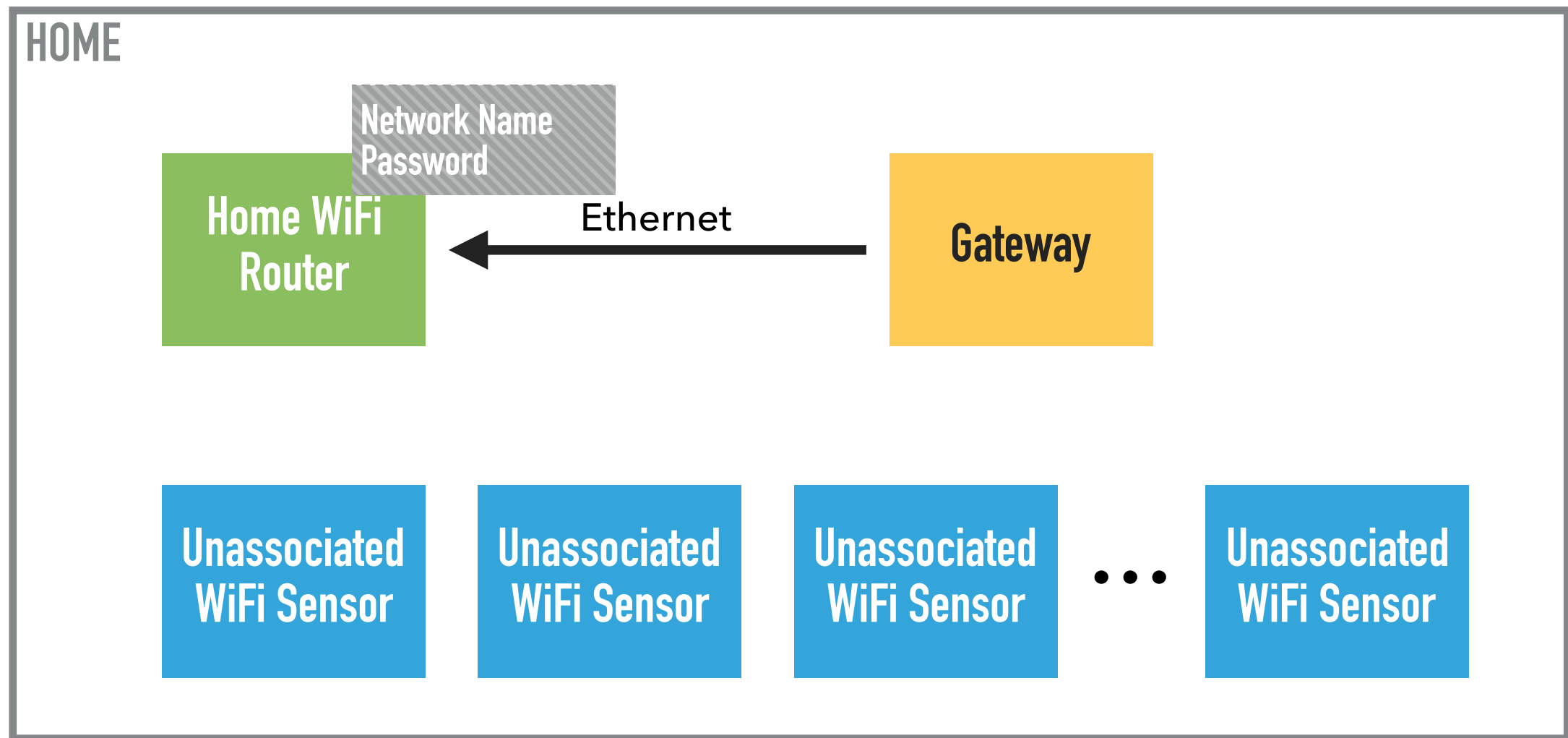
Home WiFi router securely sends network name and password to the *unassociated* sensors

STRAP



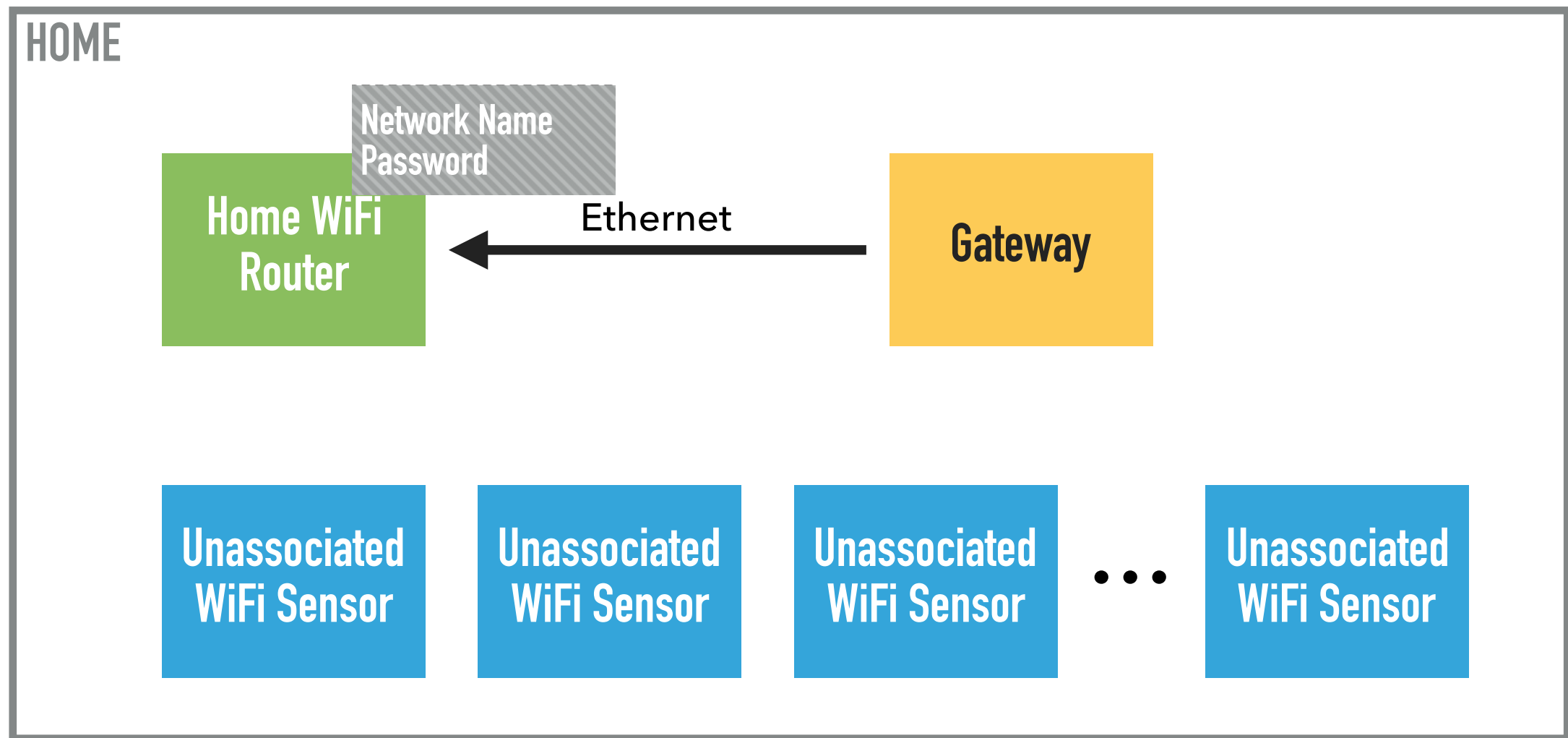
Since the gateway is connected through Ethernet, it can send data

STRAP



Since the gateway is connected through Ethernet, it can send data

STRAP PROBLEM

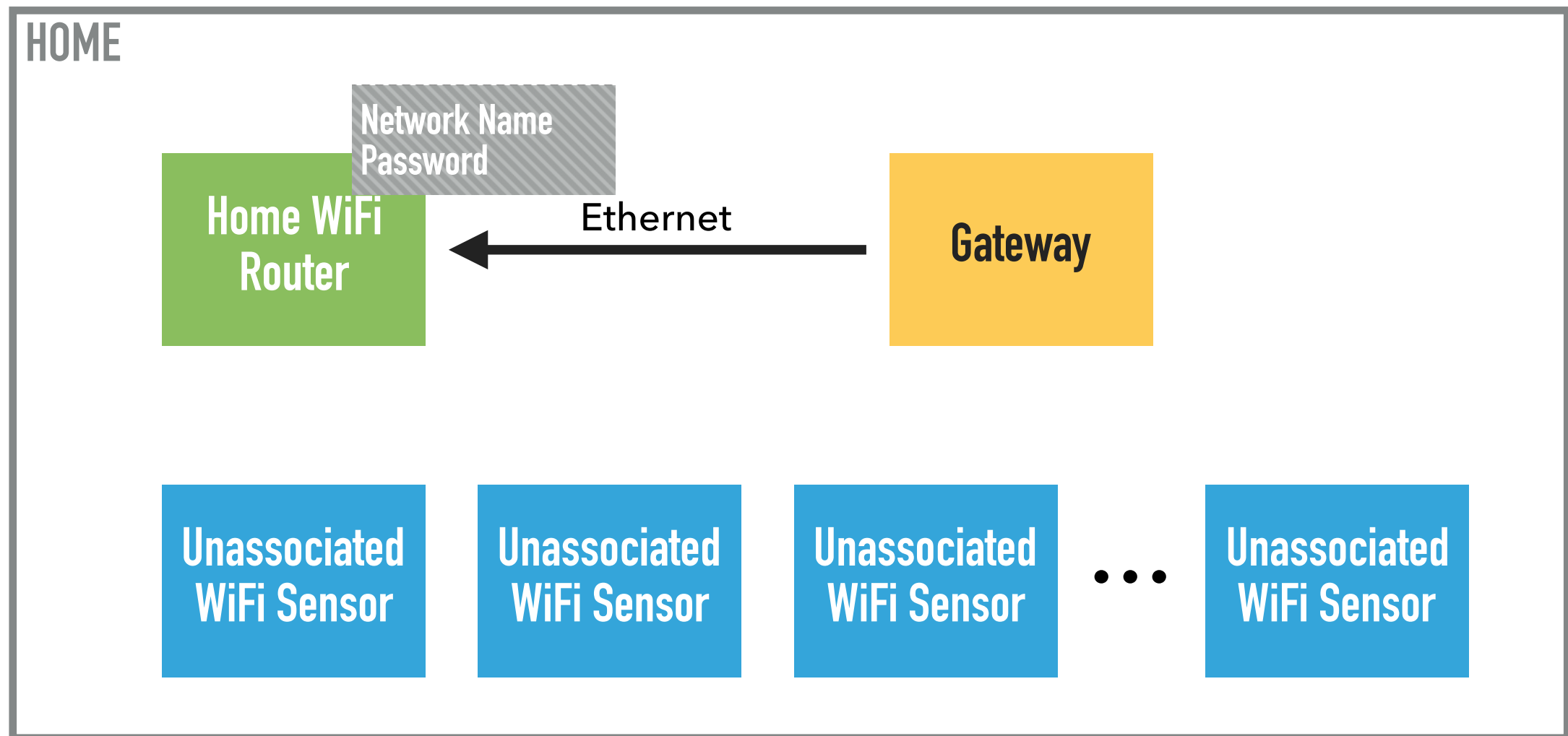


How can WiFi sensors receive WiFi frames?

MONITOR MODE

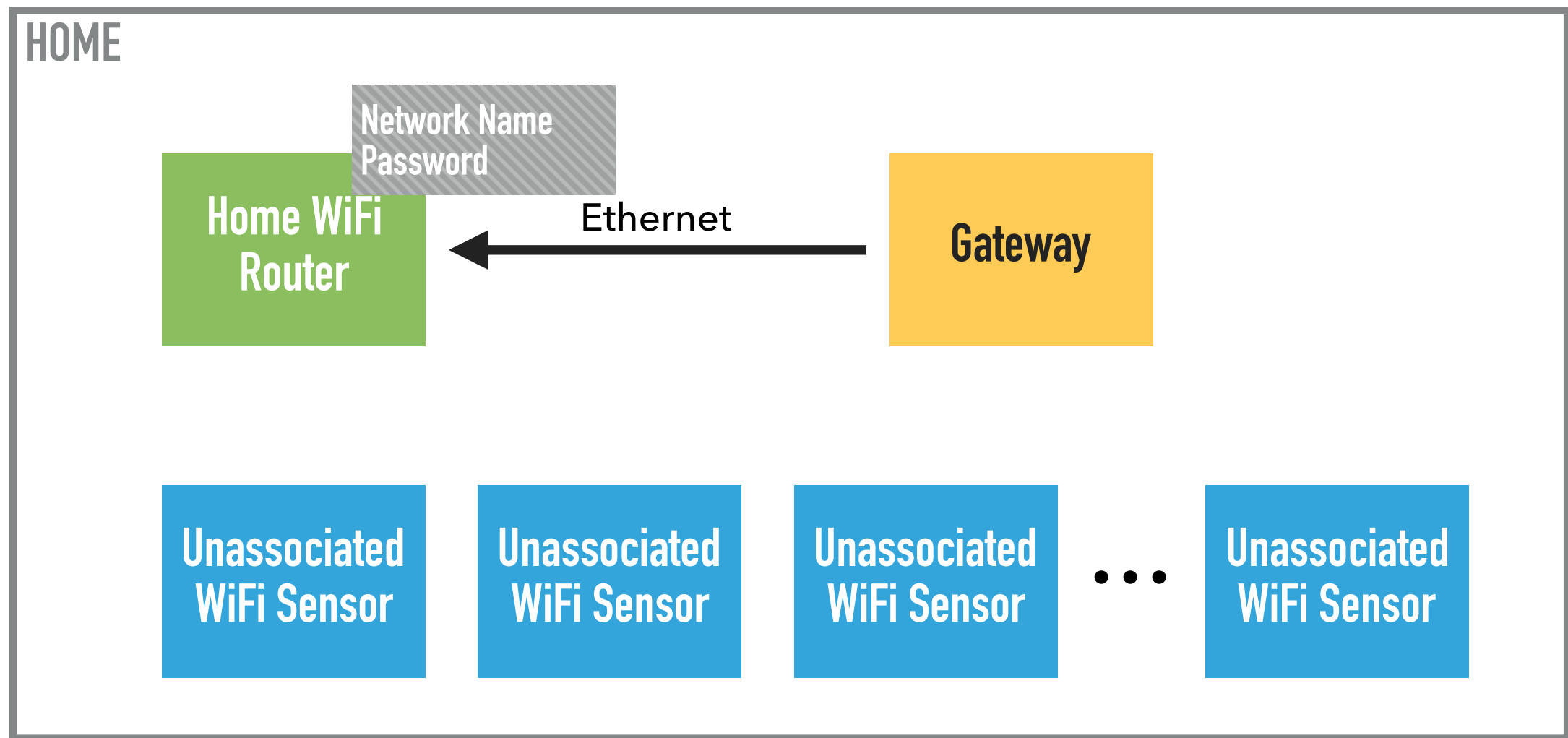
No.	Time	Source	Destination	Protocol	Data rate	Length	Info
1	0.000000	Tp-LinkT_1d:dc:1c	Broadcast	802.11	1	244	Beacon frame, SN=2497, FN=0, Flags=...
2	0.102573	Tp-LinkT_1d:dc:1c	Broadcast	802.11	1	244	Beacon frame, SN=2498, FN=0, Flags=...
3	0.138484	Mediabri_10:98:c0	Broadcast	802.11	1	254	Beacon frame, SN=589, FN=0, Flags=...
4	0.204783	Tp-LinkT_1d:dc:1c	Broadcast	802.11	1	244	Beacon frame, SN=2499, FN=0, Flags=...
5	0.240471	Mediabri_10:98:c0	Broadcast	802.11	1	254	Beacon frame, SN=591, FN=0, Flags=...
6	0.307098	Tp-LinkT_1d:dc:1c	Broadcast	802.11	1	244	Beacon frame, SN=2500, FN=0, Flags=...
7	0.388129		Tp-LinkT_1d:dc:1c (...)	802.11	24	39	Acknowledgement, Flags=.....C
8	0.388291		Tp-LinkT_1d:dc:1c (...)	802.11	24	39	Acknowledgement, Flags=.....C
9	0.389394		LiteonTe_61:ea:e5 (...)	802.11	24	39	Acknowledgement, Flags=.....C
10	0.409415	Tp-LinkT_1d:dc:1c	Broadcast	802.11	1	244	Beacon frame, SN=2501, FN=0, Flags=...
11	0.466914	2c:3a:e8:1f:69:30	Raspberr_68:93:d1	802.11	48	661	Data, SN=1519, FN=0, Flags=.p.....TC
12	0.467116		2c:3a:e8:1f:69:30 (...)	802.11	24	39	Acknowledgement, Flags=.....C
13	0.468258	Raspberr_68:93:d1	2c:3a:e8:1f:69:30	802.11	65	148	QoS Data, SN=1455, FN=0, Flags=.p.....
14	0.468437		Tp-LinkT_1d:dc:1c (...)	802.11	6	39	Acknowledgement, Flags=.....C
15	0.511608	Tp-LinkT_1d:dc:1c	Broadcast	802.11	1	244	Beacon frame, SN=2502, FN=0, Flags=...
16	0.511788	2c:3a:e8:1f:69:30	Raspberr_68:93:d1	802.11	48	117	Data, SN=1520, FN=0, Flags=.p.....TC
17	0.511942		2c:3a:e8:1f:69:30 (...)	802.11	24	39	Acknowledgement, Flags=.....C

STRAP KEY INSIGHT



Source and destination addresses of WiFi frames are *not* encrypted

STRAP SOLUTION



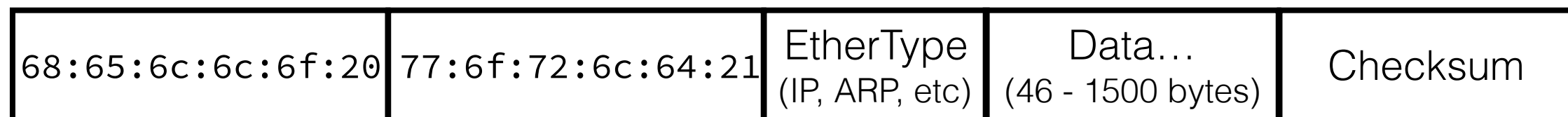
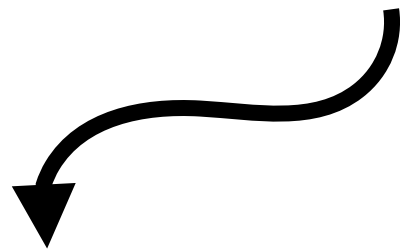
Gateway can encode data into the source and destination address fields!

ETHERNET FRAME

hello world!



68 65 6c 6c 6f 20 77 6f 72 6c 64 21

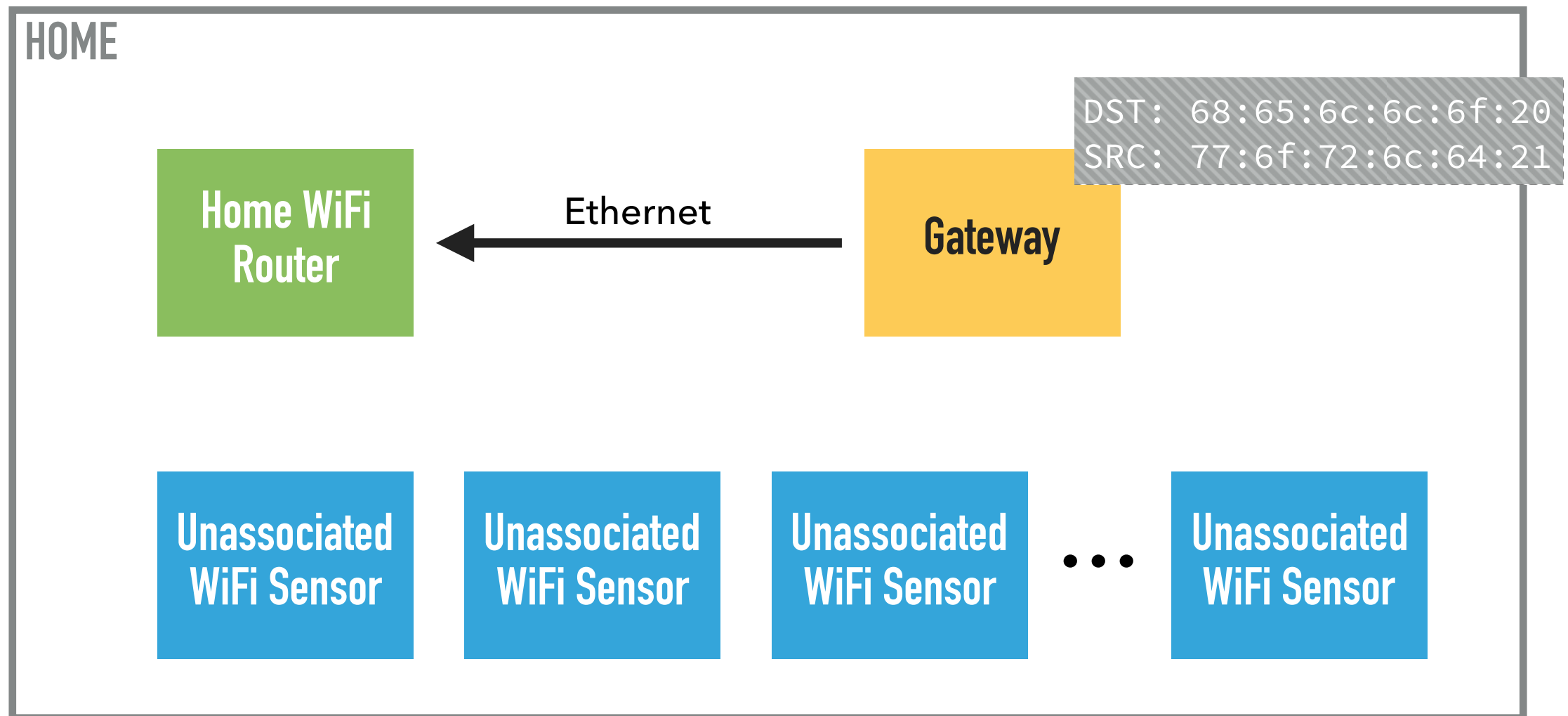


Destination MAC
Address

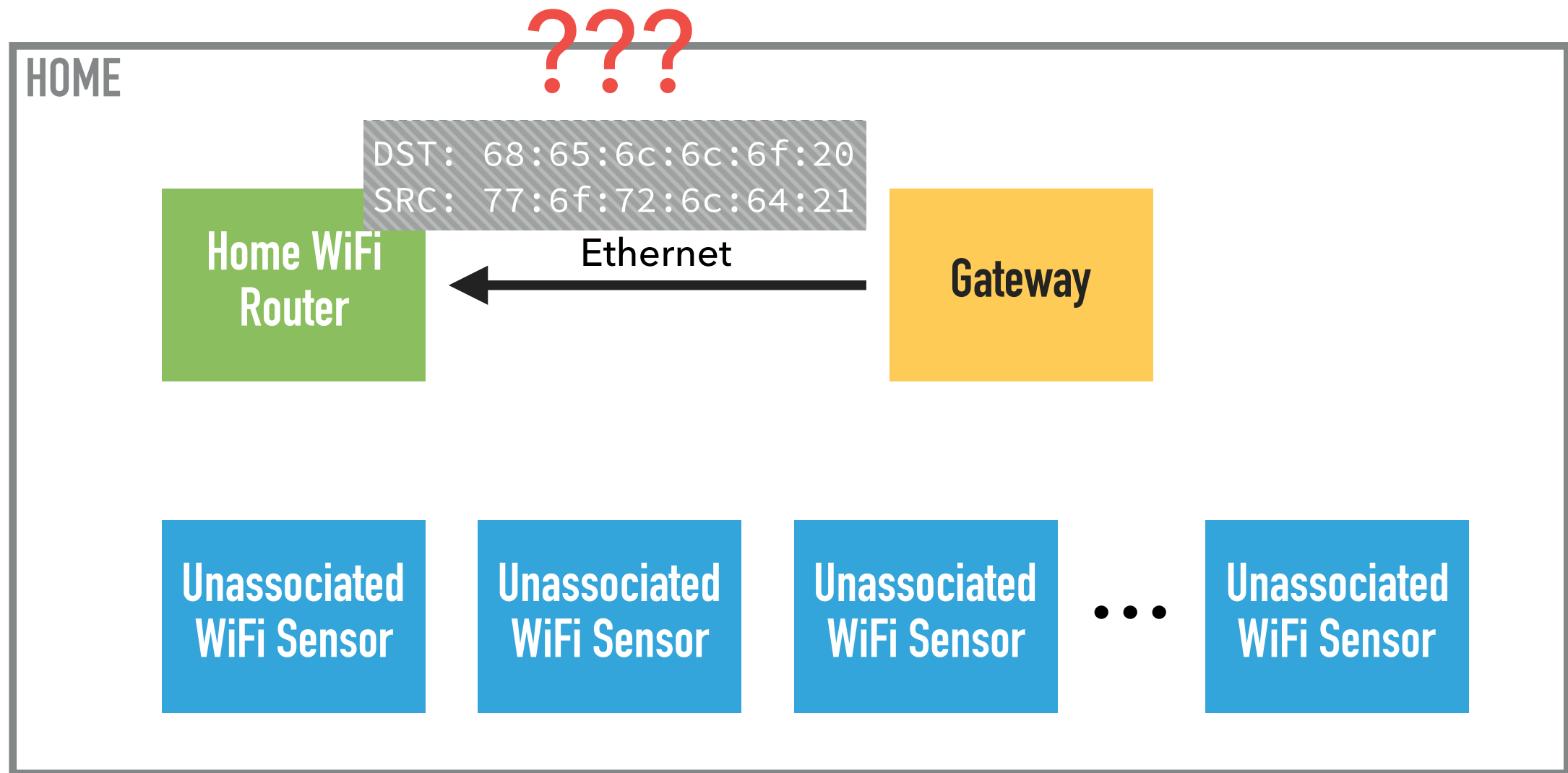
Source MAC
Address

12 bytes of data!

STRAP PROBLEM



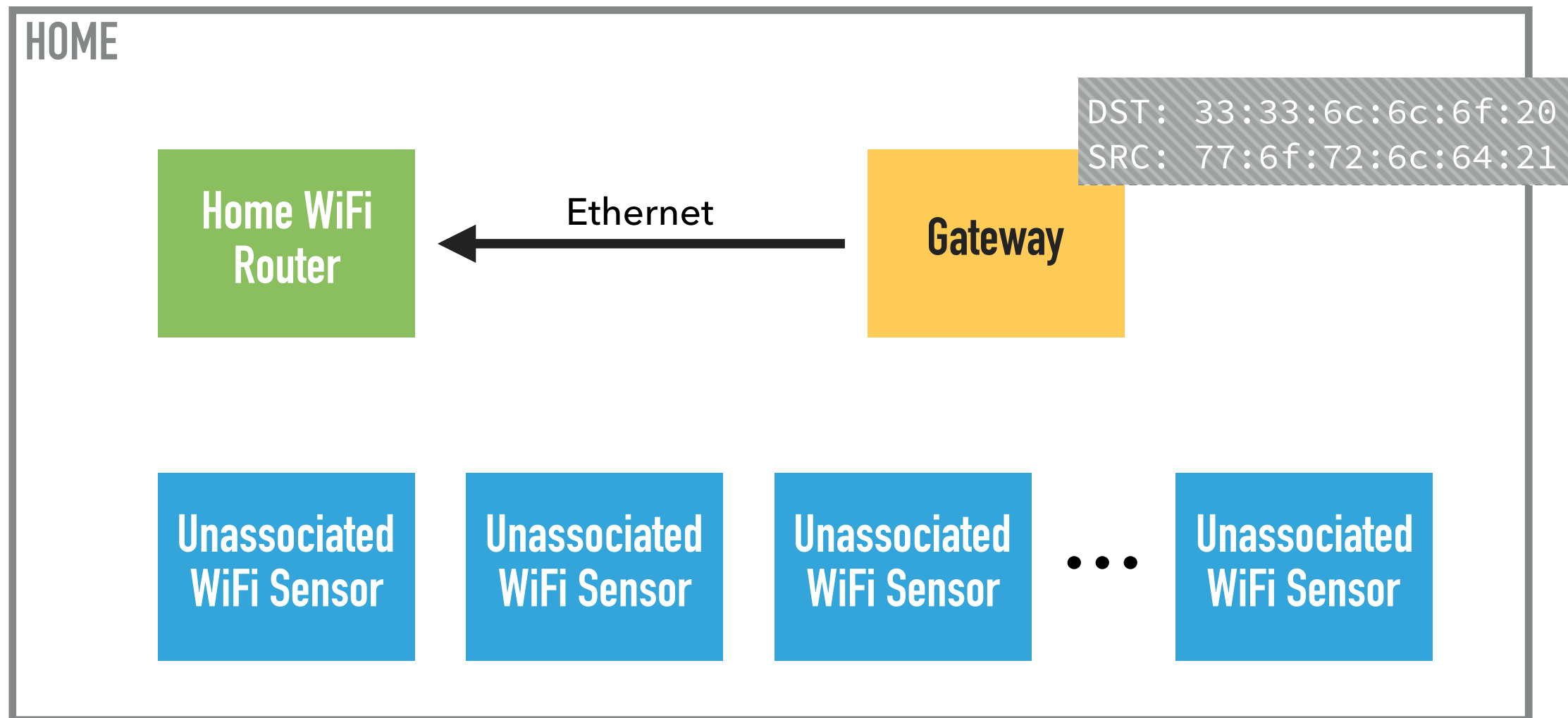
STRAP PROBLEM



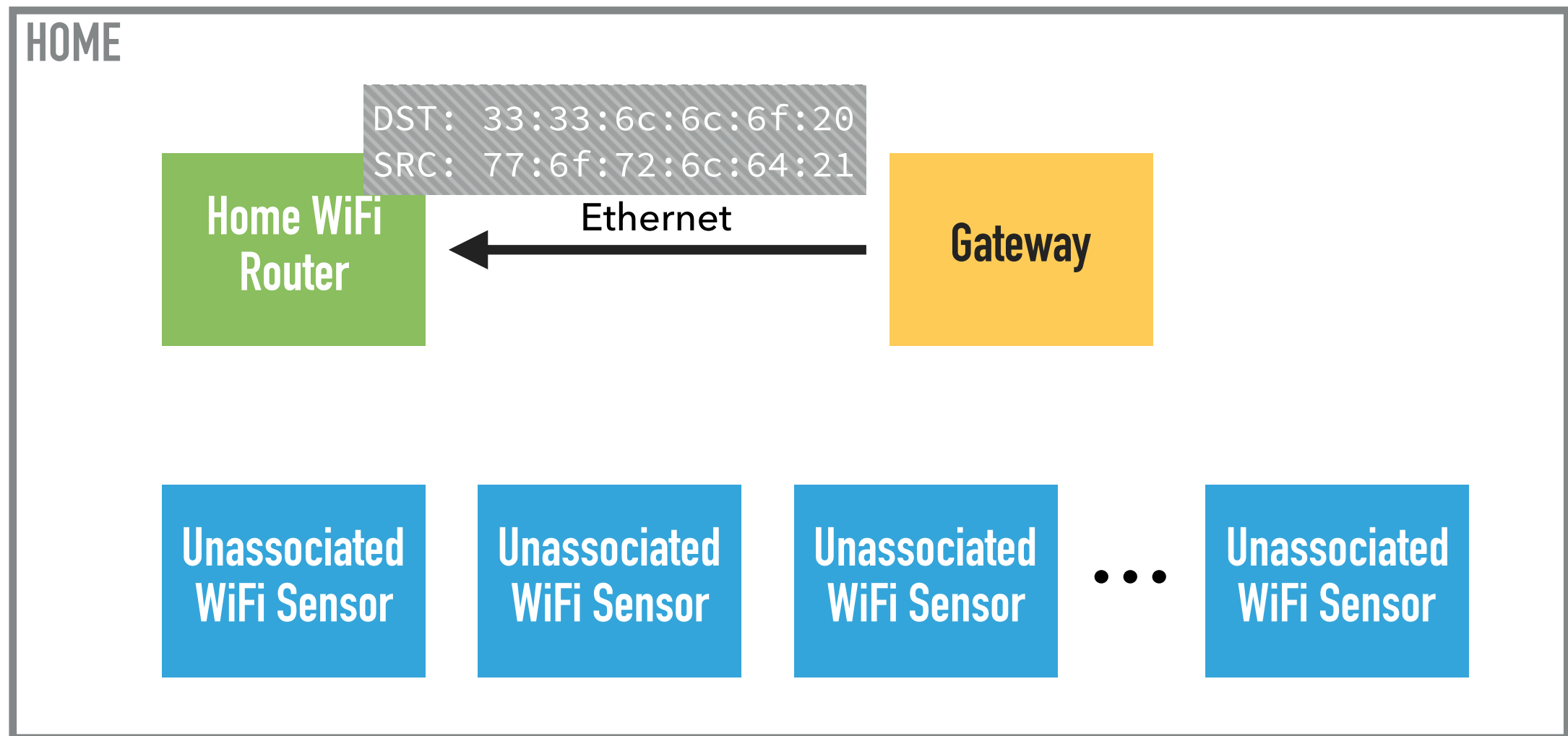
DESTINATION ADDRESS

- ▶ We need:
 - ▶ An address that the WiFi router will always accept
 - ▶ Makes the wireless router send the frame on its wireless interface
- ▶ Broadcast: ff:ff:ff:ff:ff:ff
- ▶ IPv4 Multicast: 01:00:5E:xx:xx:xx
- ▶ IPv6 Multicast: 33:33:xx:xx:xx:xx

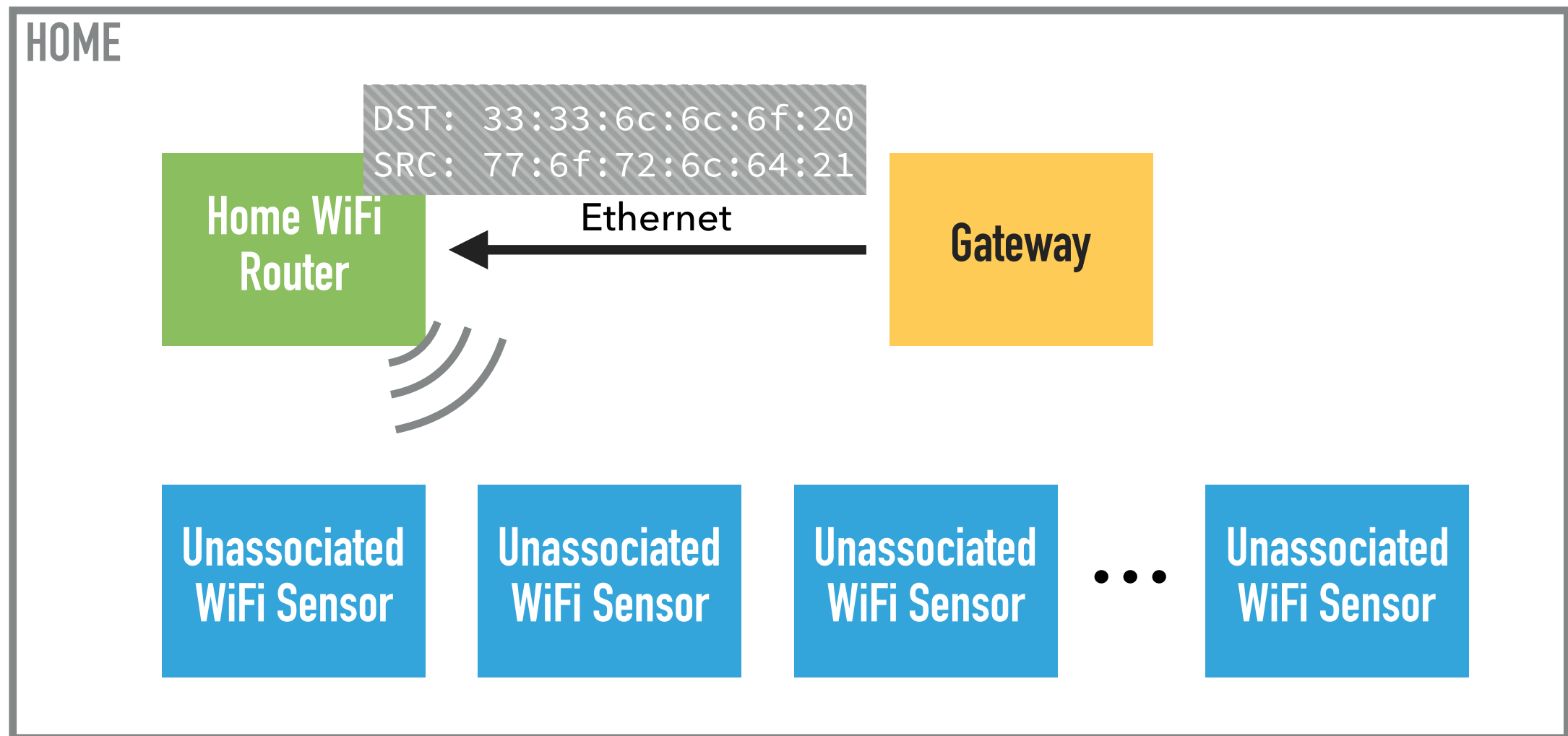
STRAP KEY INSIGHT



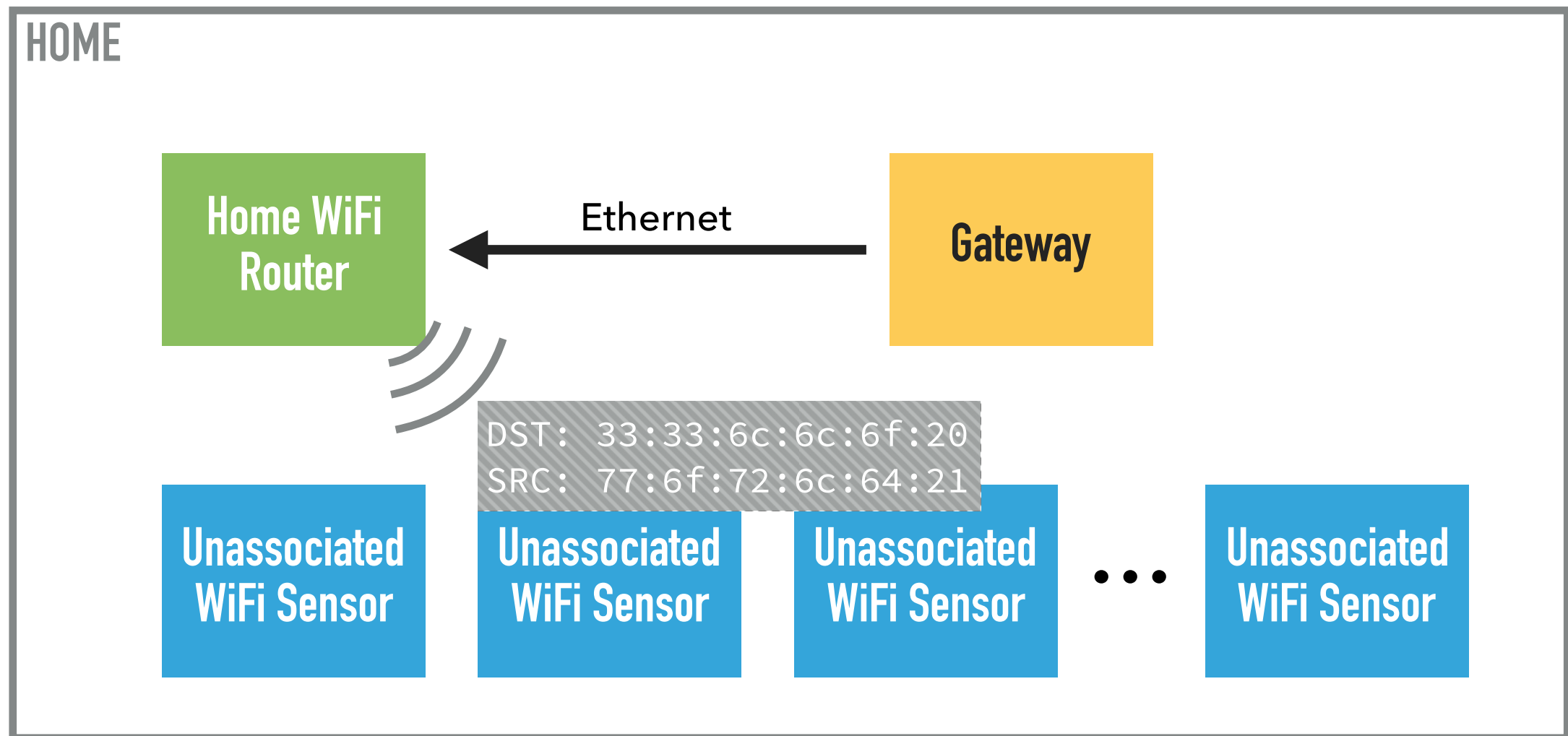
STRAP KEY INSIGHT



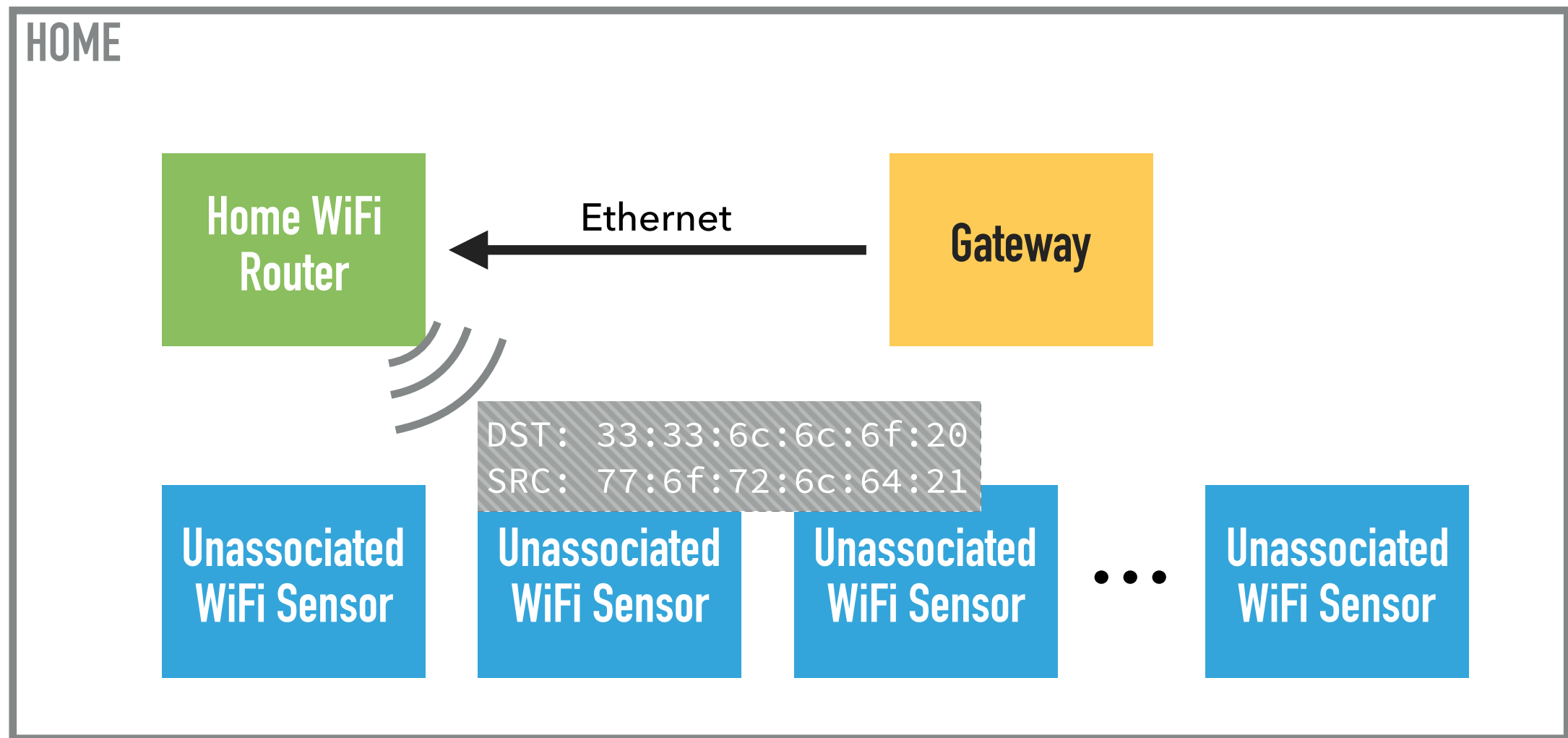
STRAP KEY INSIGHT



STRAP KEY INSIGHT

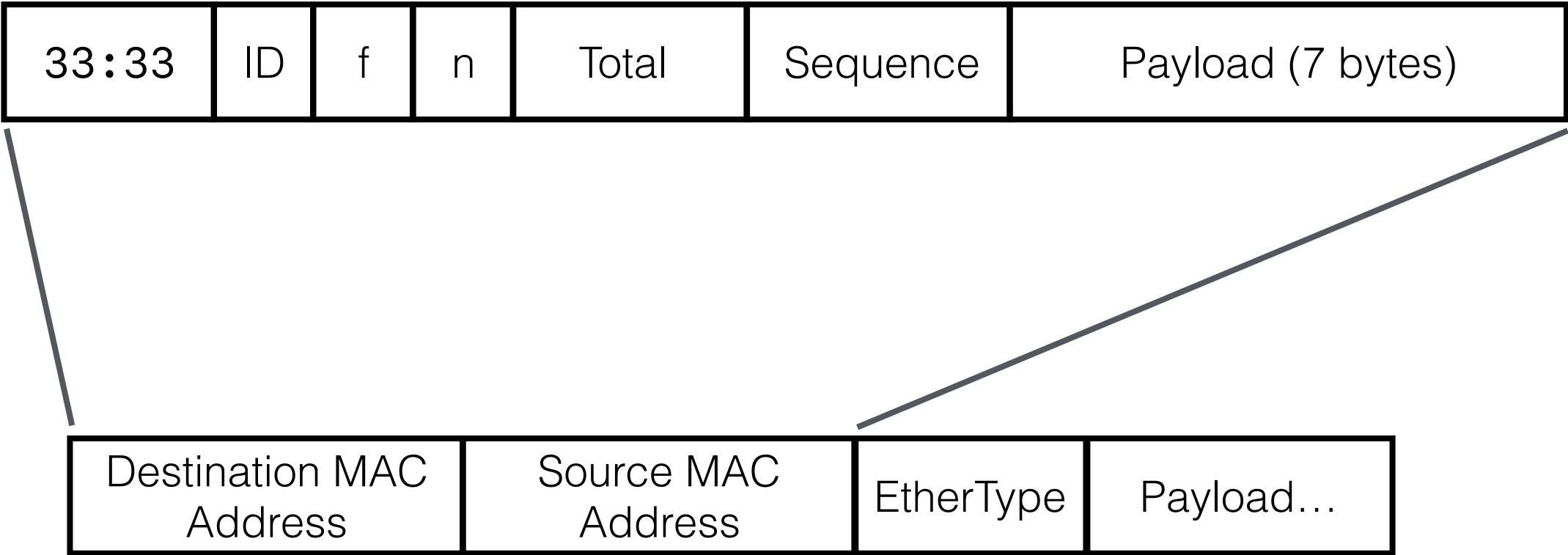


STRAP PROBLEM

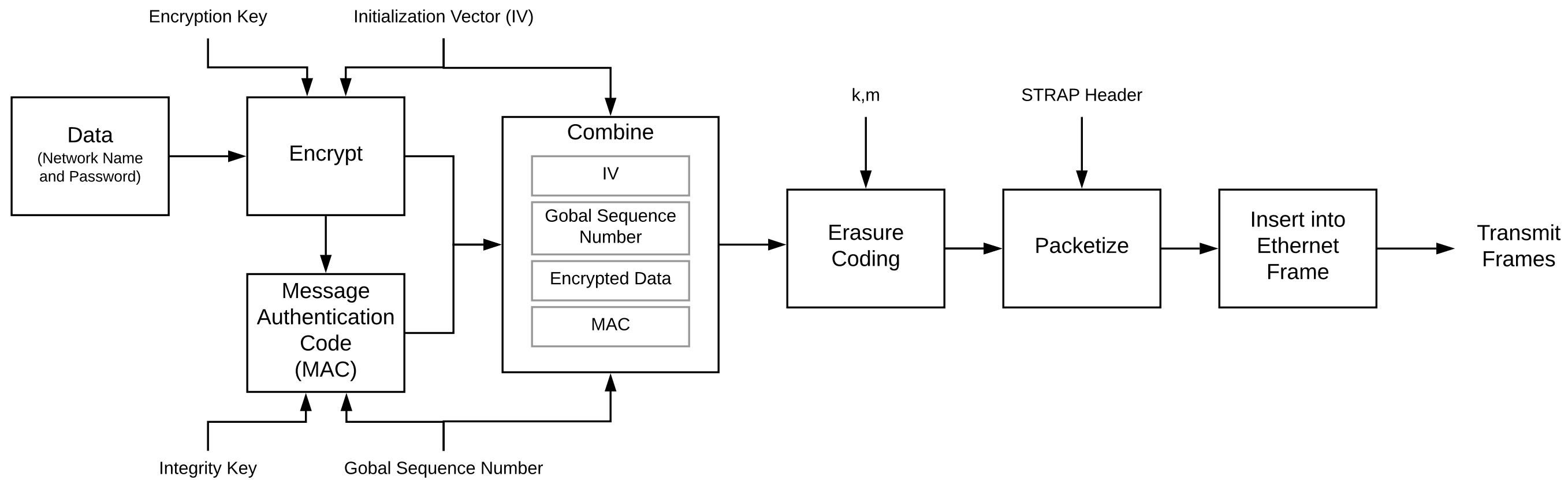


You can't fit much information in 10 bytes

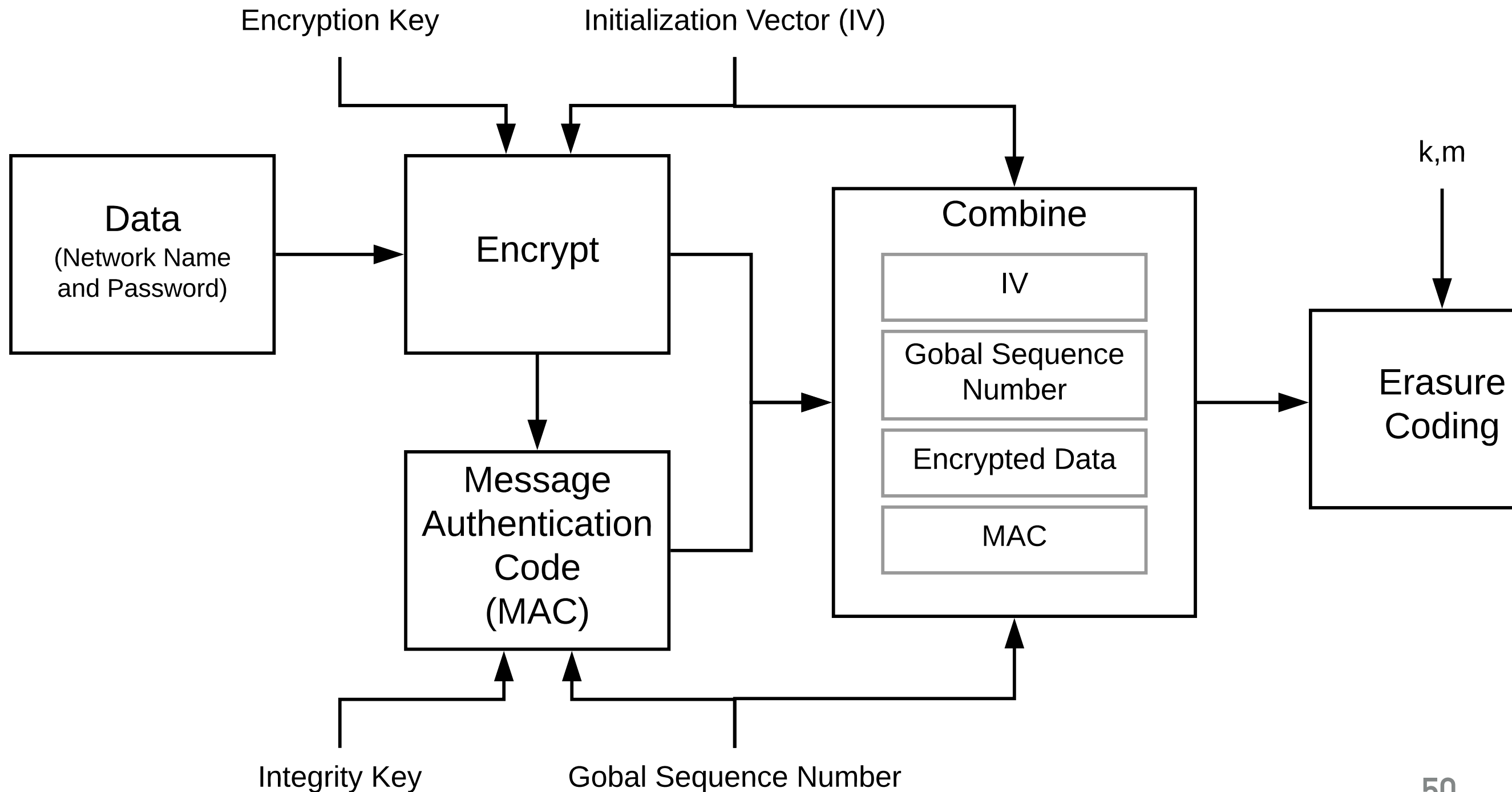
STRAP HEADER



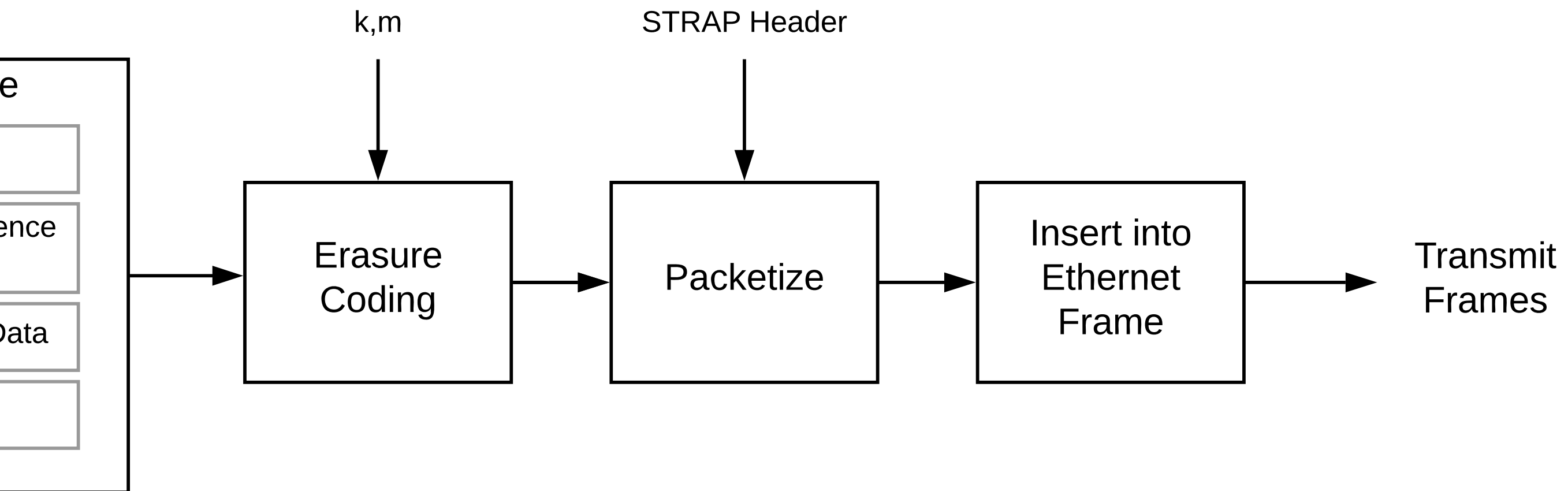
STRAP DATA FLOW



STRAP DATA FLOW



STRAP DATA FLOW



REQUIREMENTS

- ▶ Does not require any extra hardware at the sensor
- ▶ Broadly supported by home WiFi routers
- ▶ Time to connect scales up to many devices

REQUIREMENTS

- ▶ Does not require any extra hardware at the sensor
- ▶ Broadly supported by home WiFi routers
- ▶ Time to connect scales up to many devices

Uses standard WiFi to send and receive data

REQUIREMENTS

- ▶ Does not require any extra hardware at the sensor
- ▶ Broadly supported by home WiFi routers
- ▶ Time to connect scales up to many devices

Requires no changes to WiFi router

REQUIREMENTS

- ▶ Does not require any extra hardware at the sensor
- ▶ Broadly supported by home WiFi routers
- ▶ Time to connect scales up to many devices

Takes constant time, regardless of the number of sensors connecting: 1.55 seconds to 23 seconds

STRAP SUMMARY

- ▶ STRAP allows devices to send *any data* to unassociated wireless devices
- ▶ We use STRAP to send the network name and password
- ▶ Allows many devices to connect at once in *constant time* without requiring any extra hardware

PROBLEMS ENCOUNTERED

- ▶ Having other people use our system lead to many interesting challenges:
 - ▶ Managing deployments
 - ▶ Bootstrapping WiFi connectivity
 - ▶ **Device observability**
 - ▶ Data privacy when moving sensors
- Solved**
- Future work**

DEVICE OBSERVABILITY MOTIVATION

- ▶ How can a remote manager know if a WiFi device is functioning or not?
- ▶ Problem is especially bad for difficult-to-access locations

Deployment 007

Monitor Name	Online	Last received PM Data
monitorb003	<div></div>	3 days
monitorb018	<div></div>	1 minute
monitorb016	<div></div>	5 hours

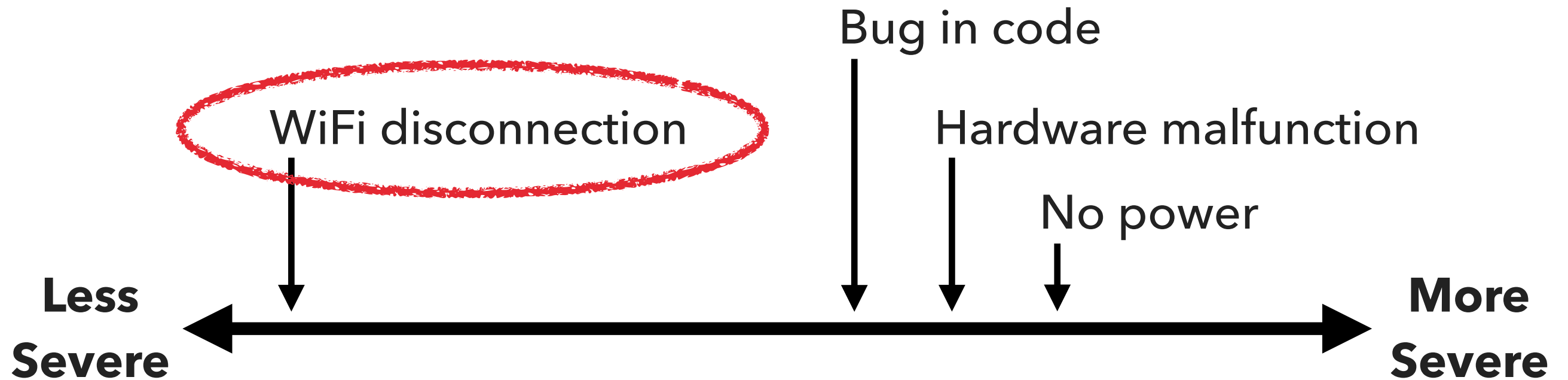
Deployment 008

Monitor Name	Online	Last received PM Data
monitorb019	<div></div>	1 minute
monitorb009	<div></div>	40 seconds
monitor110	<div></div>	57 seconds

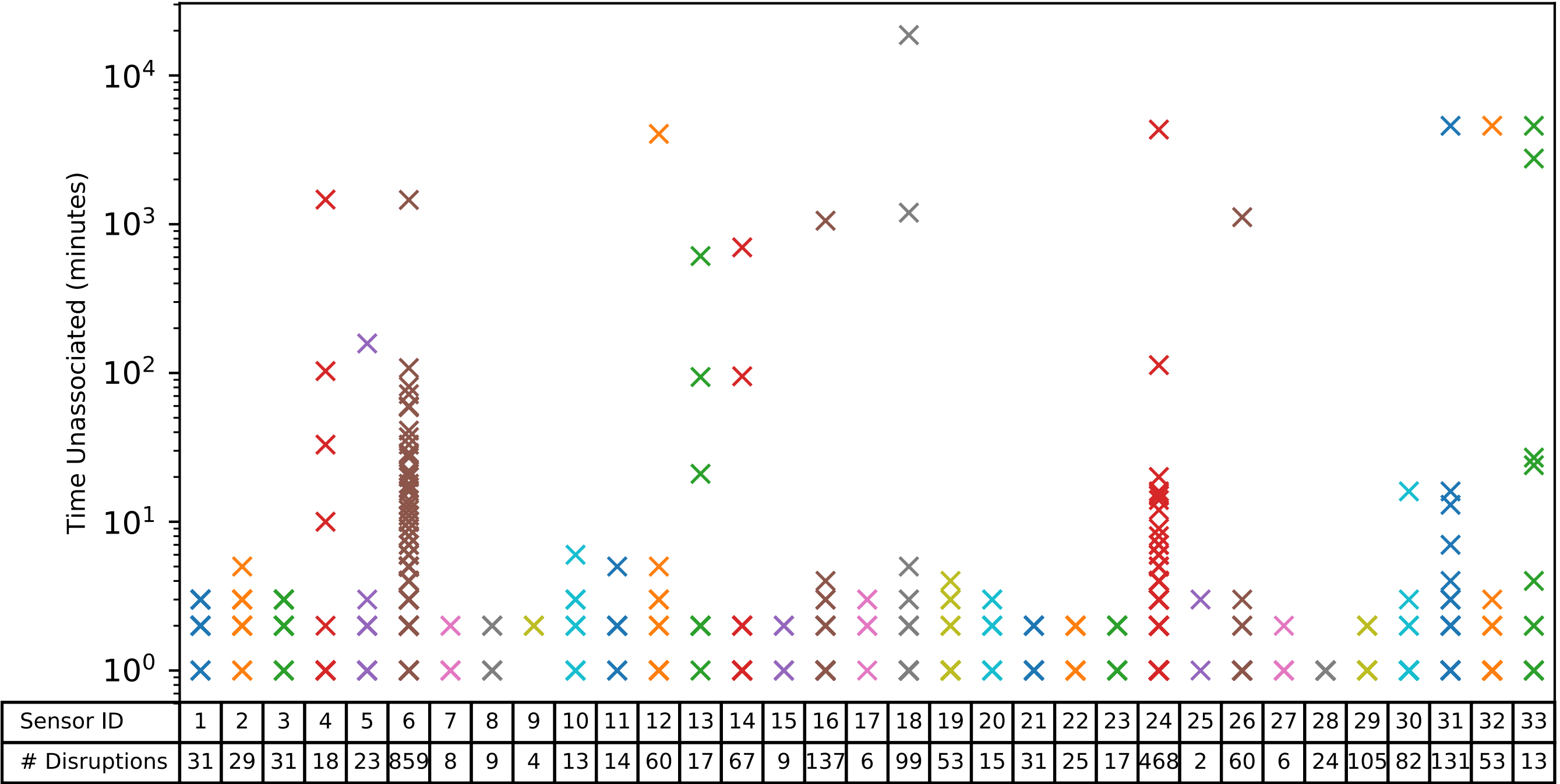
Deployment 009

Monitor Name	Online	Last received PM Data
monitorb001	<div></div>	41 seconds
monitorb002	<div></div>	1 minute
monitorb014	<div></div>	47 seconds

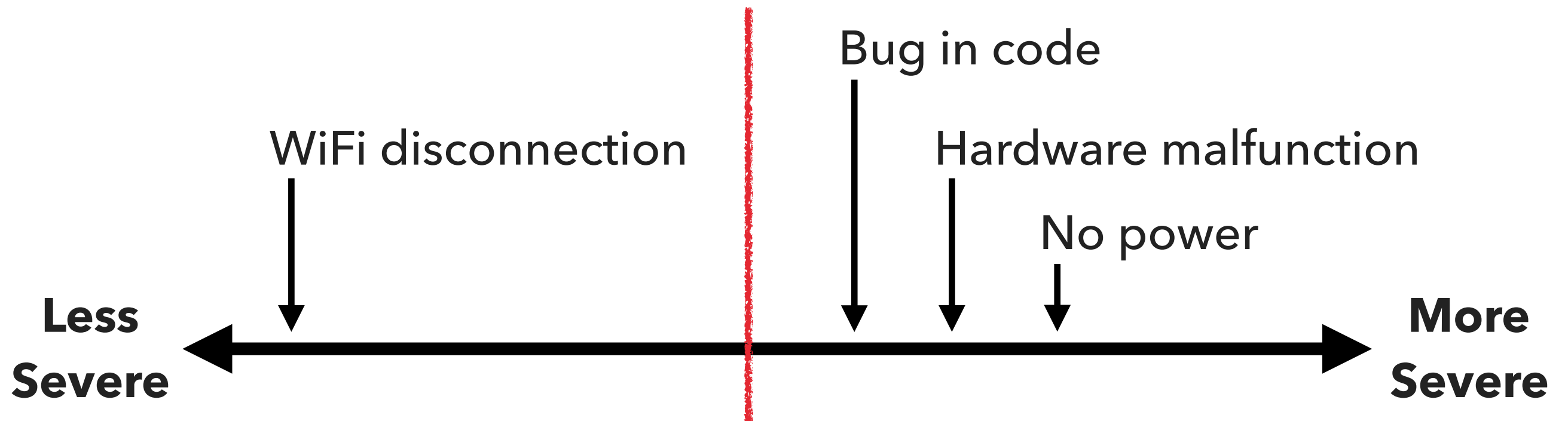
DISRUPTION TYPES



WIFI DISRUPTIONS



DISRUPTION TYPES



Distinguish between WiFi disruptions and other types of disruptions

LONGER RANGE COMMUNICATIONS COMPARED TO WIFI

- ▶ Cellular
- ▶ LoRa
- ▶ 802.11ah (HaLow)
- ▶ All solutions require different radios

PROBLEMS ENCOUNTERED

- ▶ Having other people use our system lead to many interesting challenges:
 - ▶ Managing deployments
 - ▶ Bootstrapping WiFi connectivity
 - ▶ Device observability
 - ▶ **Data privacy when moving sensors**
- Solved**
- Future work**

DATA PRIVACY

- ▶ Sensor's location is managed by metadata
- ▶ If metadata is not updated, system will attribute data to wrong location
- ▶ Sensors should learn location and detect when the location has changed

SUMMARY

- ▶ We built EpiFi to make deploying and running epidemiological studies easier
- ▶ While building and deploying system, we ran into many challenges
- ▶ We solve management issues by building tools to view sensor status and export sensor data
- ▶ We solve the WiFi bootstrapping problem with STRAP
- ▶ Many more challenges that need to be solved!
- ▶ <https://github.com/VDL-PRISM>

Questions