B6:The Ragtag Tag Team

Udit Ranasaria, Shiva Nathan, Rhea Kudtarkar

Application Area

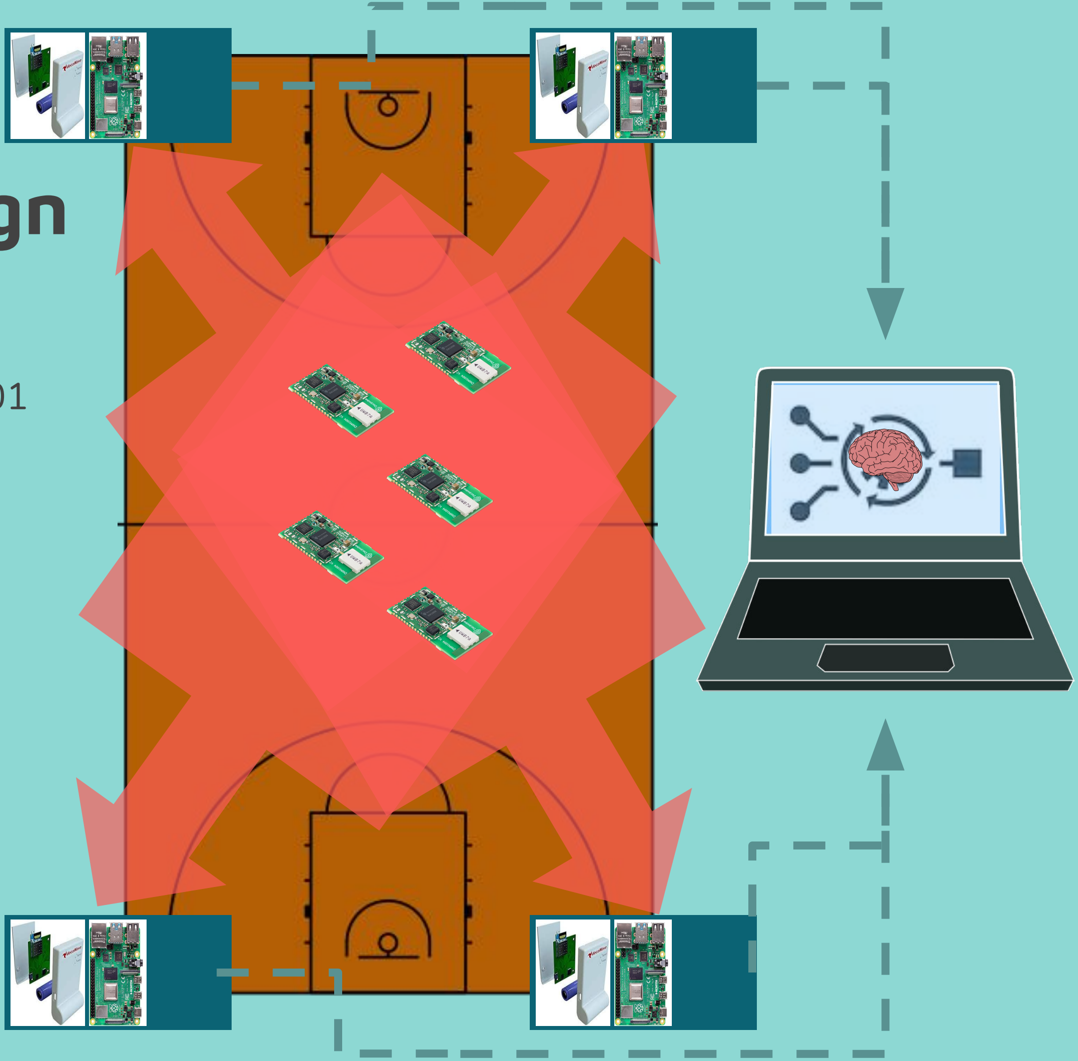
- Affordable Real-Time Location System for NCAA Basketball
- Records locations of each player on the court into a database
- NBA “Advanced Stats”

Solution Approach

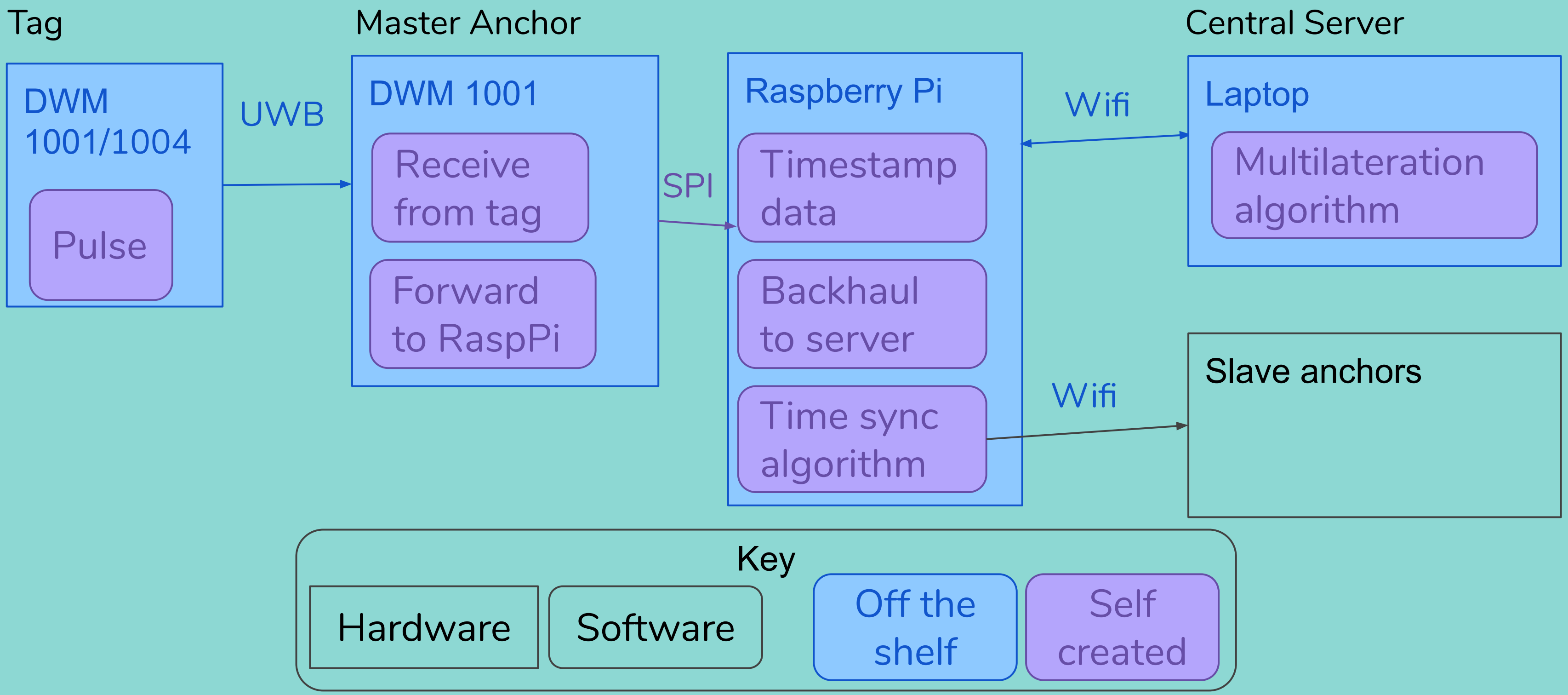
TDOA Concept	TDOA Implementation
Tag emits pulse with its ID	Decawave chips programmed using the API
Anchors around court receive pulse & log time of reception	Decawave chips send data to Raspberry Pi, which timestamps it using it's internal clock
Anchors transmit info to central server	Raspberry Pi transmits timestamped data to a computer over wifi
Server determines location of tag based on time difference	Computer runs multilateration algorithm

Solution Approach: Design

- **Tags:** Decawave DWM1004/1001
- **Anchors:** DWM1001 + RaspPi
- **Server:** Laptop + Algorithm
- **Connectivity:**
  - UWB from tags to anchors
  - Wi-Fi rom anchors to server



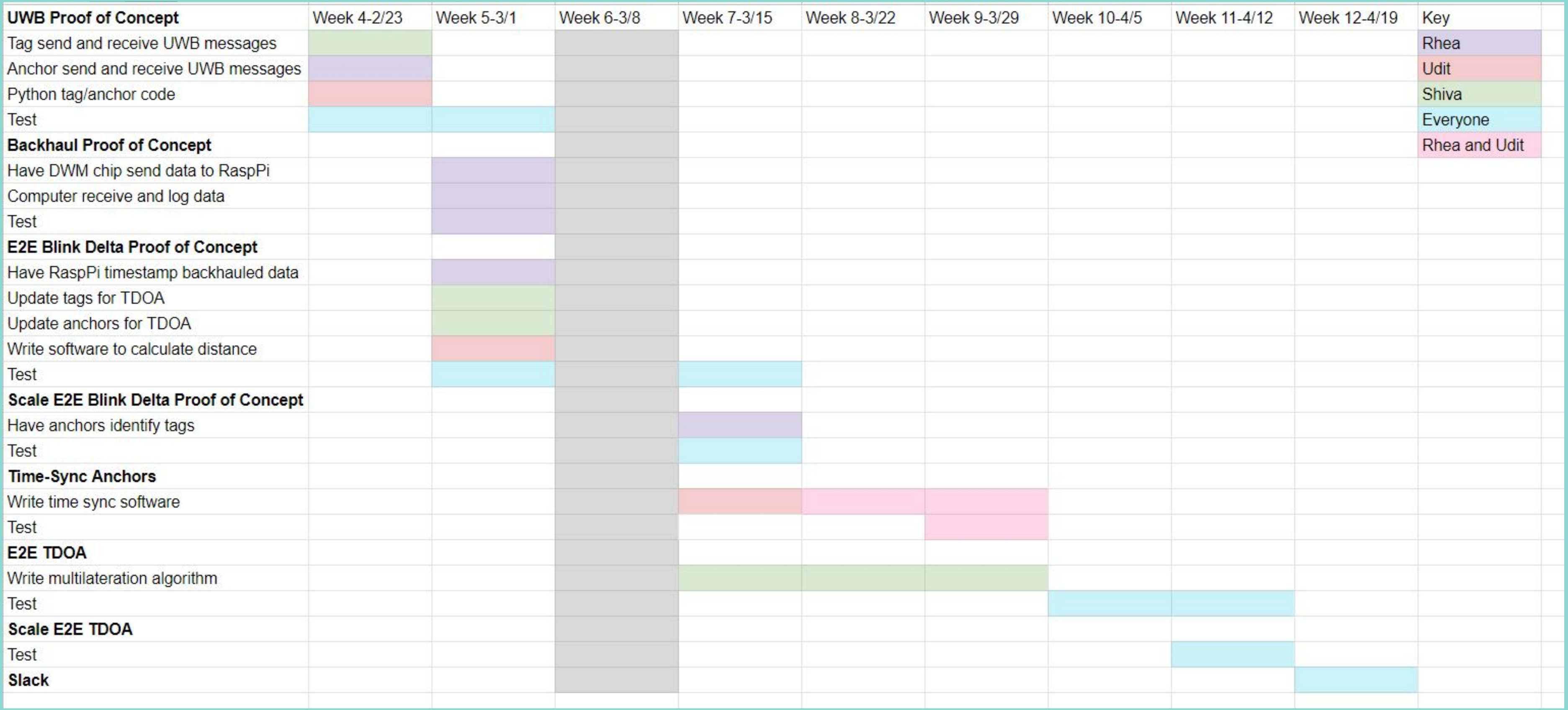
System Specification / Block Diagram



Implementation Plan

- Use pre-owned DWM1001 UWB chips as Tags and Anchors for early work
- Interface with ECE RasPis3/4s for backhaul and faster clocks
- Use our own laptops for server computation
- Ansys for large room testing
- Will upgrade to smaller/more test-friendly DW1004 UWB chips if needed

Project Management



UWB Proof of Concept	Week 4-2/23	Week 5-3/1	Week 6-3/8	Week 7-3/15	Week 8-3/22	Week 9-3/29	Week 10-4/5	Week 11-4/12	Week 12-4/19	Key
Tag send and receive UWB messages	🟩		▒							🟪 Rhea
Anchor send and receive UWB messages	🟪		▒							🟥 Udit
Python tag/anchor code	🟥		▒							🟩 Shiva
Test	🟦	🟦	▒							🟦 Everyone
Backhaul Proof of Concept			▒							🩷 Rhea and Udit
Have DWM chip send data to RaspPi		🟪	▒							
Computer receive and log data		🟪	▒							
Test		🟪	▒							
E2E Blink Delta Proof of Concept			▒							
Have RaspPi timestamp backhauled data		🟪	▒							
Update tags for TDOA		🟩	▒							
Update anchors for TDOA		🟩	▒							
Write software to calculate distance		🟥	▒							
Test		🟦	▒	🟦						
Scale E2E Blink Delta Proof of Concept			▒							
Have anchors identify tags			▒	🟪						
Test			▒	🟦						
Time-Sync Anchors			▒							
Write time sync software			▒	🟥	🩷	🩷				
Test			▒			🩷				
E2E TDOA			▒							
Write multilateration algorithm			▒	🟩	🟩	🟩				
Test			▒				🟦	🟦		
Scale E2E TDOA			▒							
Test			▒					🟦		
Slack			▒						🟦	

Test Metrics

Test Name	Methodology (input)	Reasoning	Specs(success)
UWB Proof-of-concept	1 mobile-tag, 1 fixed-anchor. Use ToF to calculate distance in straight line.	Prove UWB chip programming and viability of using clocks in the first place	Up to 35 meters range, 0.5 meter accuracy
Backhaul proof-of-concept	Same test as above but data is backhauled to server	Prove viability of backhaul mechanism (interface UWB chip with RaspPi)	No dropped packets/info
E2E Blink Delta Proof-of-concept	1 mobile-tag, 1 fixed-anchor. Use TDoA to calculate distance deltas in straight line.	Validate E2E TDoA process without multiple anchors/tags and complex software. Have to add timestamps to blinks along the way and server does math now	Up to 35 meters range, 0.5 meter accuracy
Scale E2E Blink Proof-of-concept	Scale to 4 anchors.	Scale	No data loss and above TDoA metrics
Time-sync anchors	1 fixed-tag, 4 fixed anchors equidistant from tag. Analyze difference in received timestamps at anchors	Test and improve timestamping drift and difference. Need to try many options if doesn't work	<2ns (translates to foot level accuracy as requirements said)
E2E TDoA	1 tag, 4 fixed anchors	Add TDoA multi-lat algos to get absolute position	1 foot
Scale E2E TDoA	10 tag, 4 fixed anchors	Final proj validation!	ditto

Handling Failure & Risk in

Test Name	Risks/Tech issues	Failure Contingencies
UWB Proof-of-concept	● Custom programming of chips ● UWB radios don't suffice	● Make our own chips with another radio system ● Remove UWB entirely
Backhaul proof-of-concept	● SPI interface doesn't cooperate	● Do calculations on RPi ● Find new backhaul mechanism
E2E Blink Delta Proof-of-concept	● Custom programming of chips ● UWB radios don't suffice	● Make our own chips with another radio system ● Remove UWB entirely
Scale E2E Blink Proof-of-concept	● Backhaul (wifi, ble) caves under load ● UWB interference	● Find stronger radios ● Ditch UWB
Time-sync anchors	● Clocks just aren't consistent ● High drift ● Computation differences are too high	● Use TCXOs ● Use wired time sync with a single TCXO clock
E2E TDoA	● Multi-lat algos too complex ● Computation too expensive ● Incoming data is not coming is not good enough	● Read papers and find alternatives to algo ● Switch to TWR ● Push comp to cloud
Scale E2E TDoA	● Backhaul (wifi, ble) caves under load ● UWB interference	● Scale back scaling goals