UMASS AMHERST

SDP 18: EfficienSeat

Midway Design Review
5 December 2017

Team 26
Advised by Professor Tilman Wolf
Kristina Georgadarellis, Matthew Donnelly,
Dennis Donoghue, Aarsh Jain

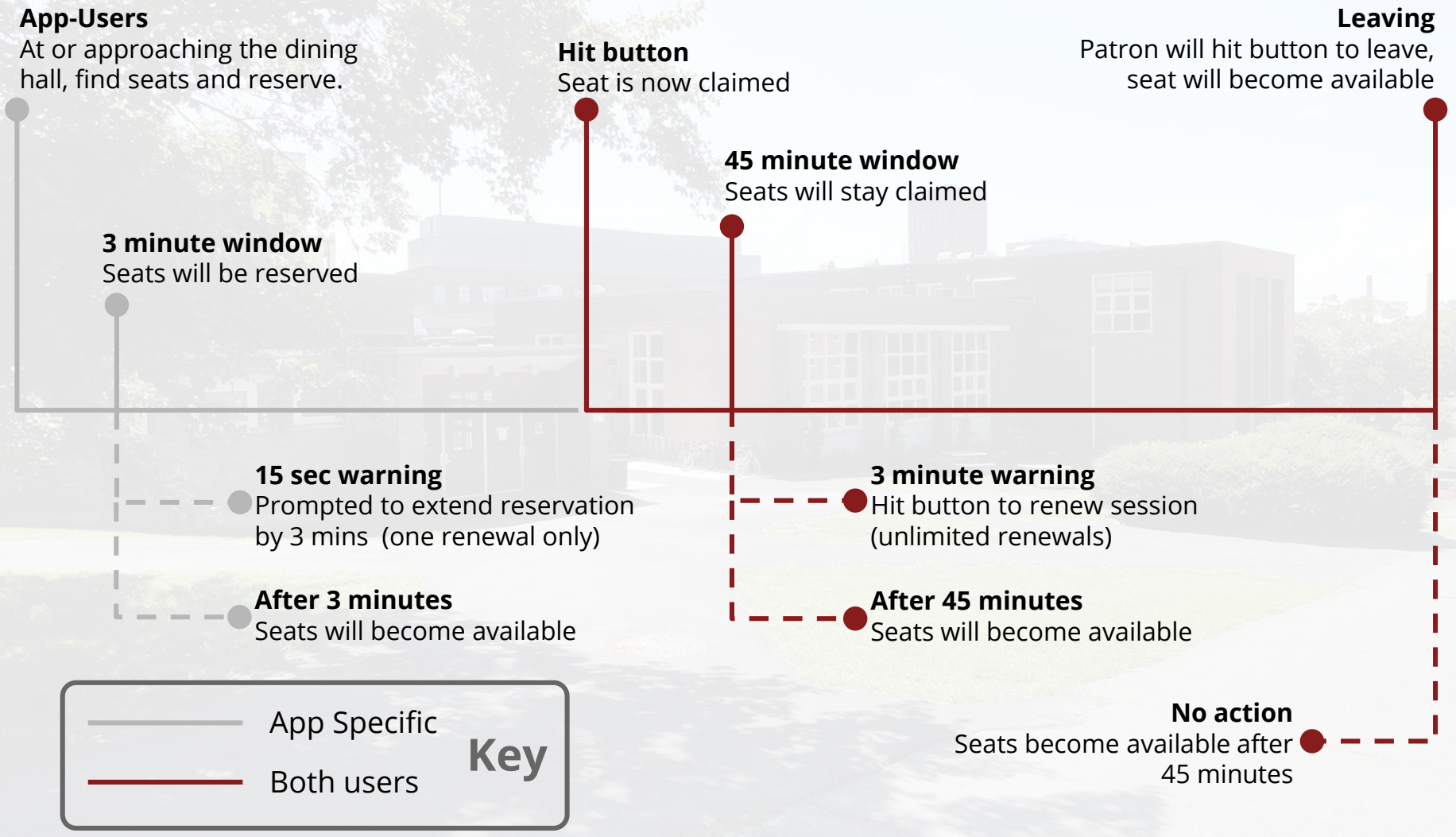
Problem Statement

- Dining halls suffer from inefficiencies due to their current method of seating patrons
- A potential solution addresses needs of the patrons as well as the dining hall
  - Patrons need an easy way to identify and locate open seats in dining halls
  - Dining halls need an easily implementable and maintainable system

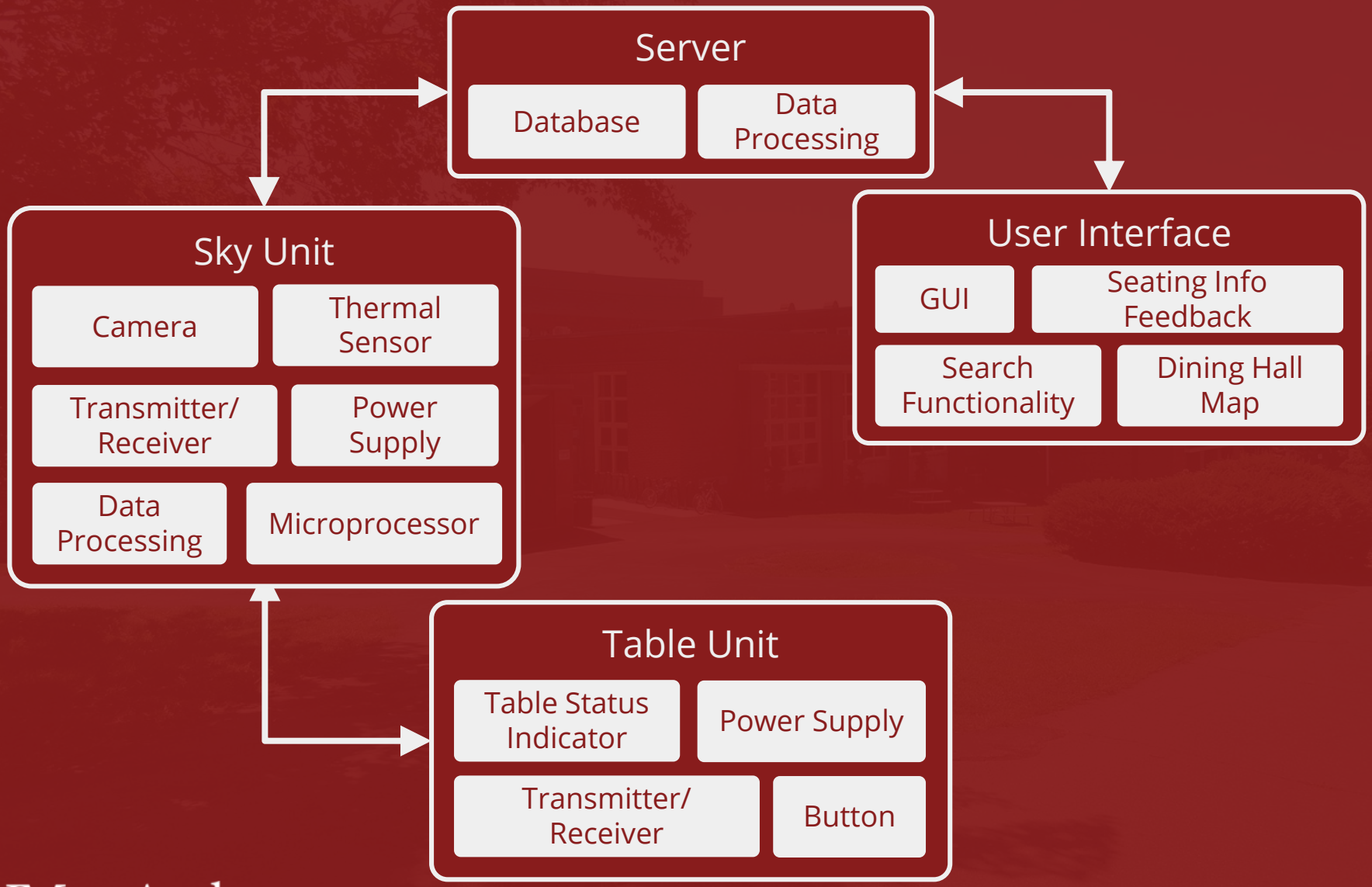
System Specifications

- Allows app users to search for available seats by party size through app
- Users will receive response from the app within 2 seconds
- The app will generate a depiction of table locations accurate to ½ a table length
- Can function in a dining hall of several hundred seats
- Table unit is compliant with IPX4
- System accommodates non-app using patrons
- Needs to be maintained once a month

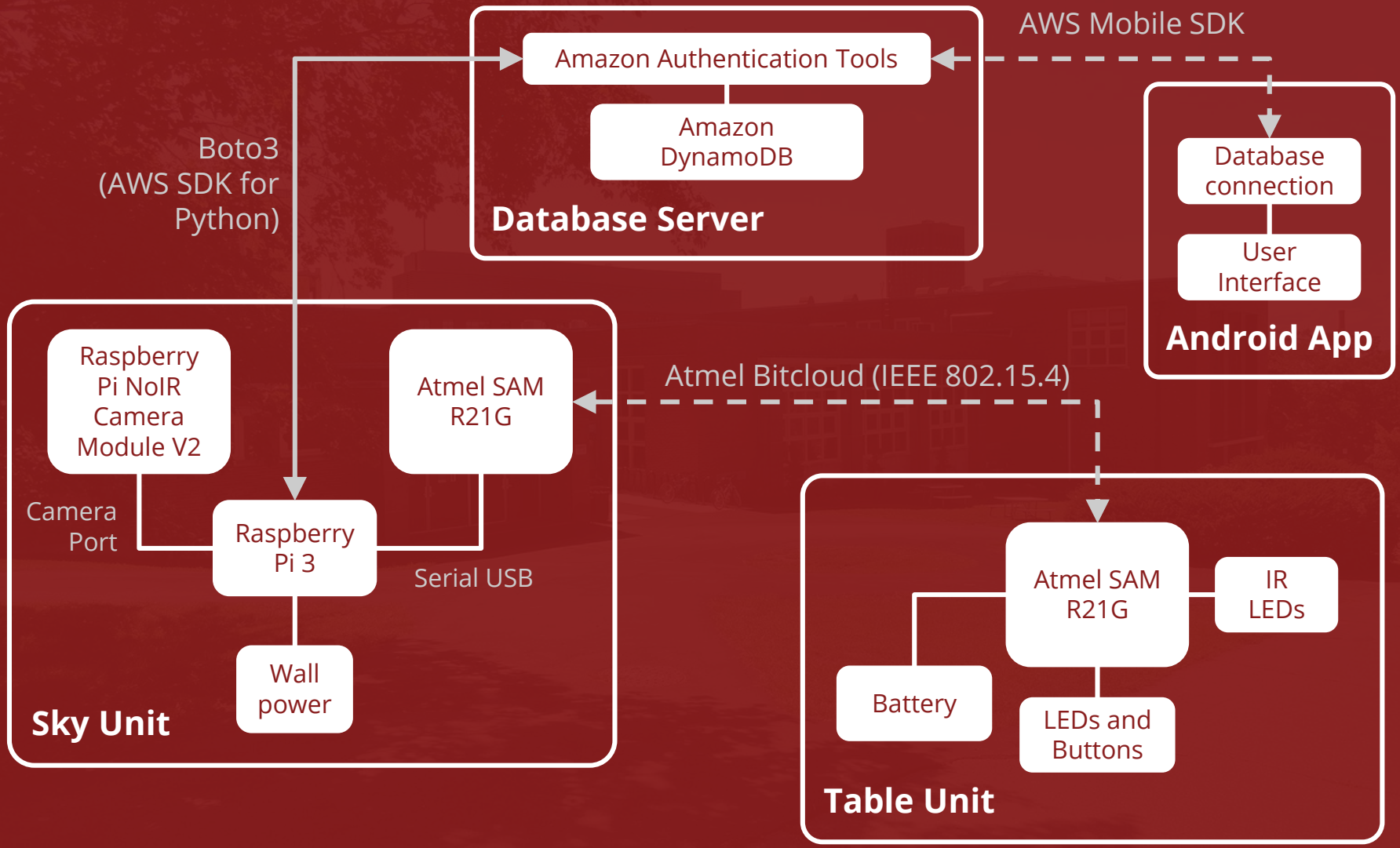
Timeline for User Interaction



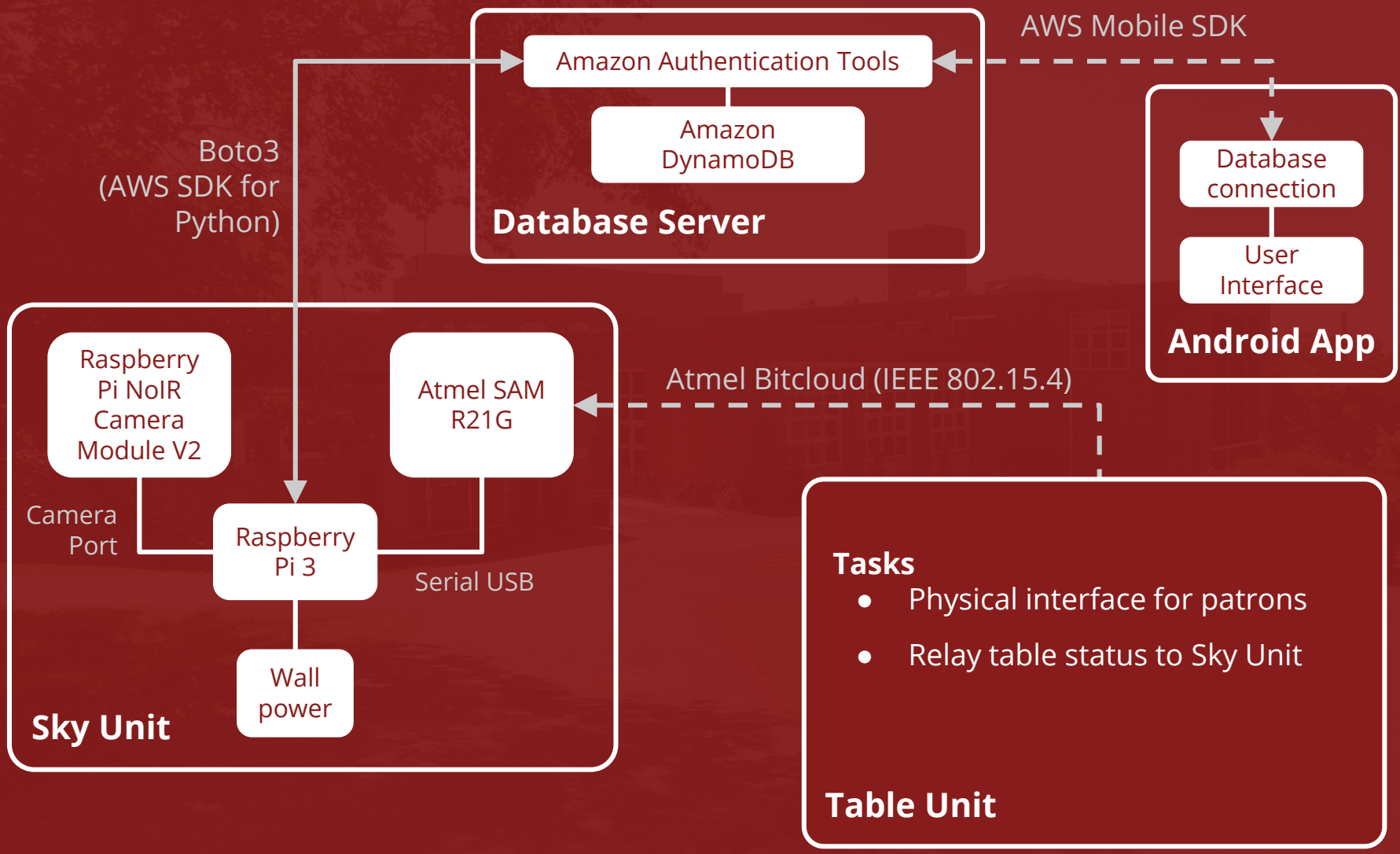
Previous Solution: Block Diagram



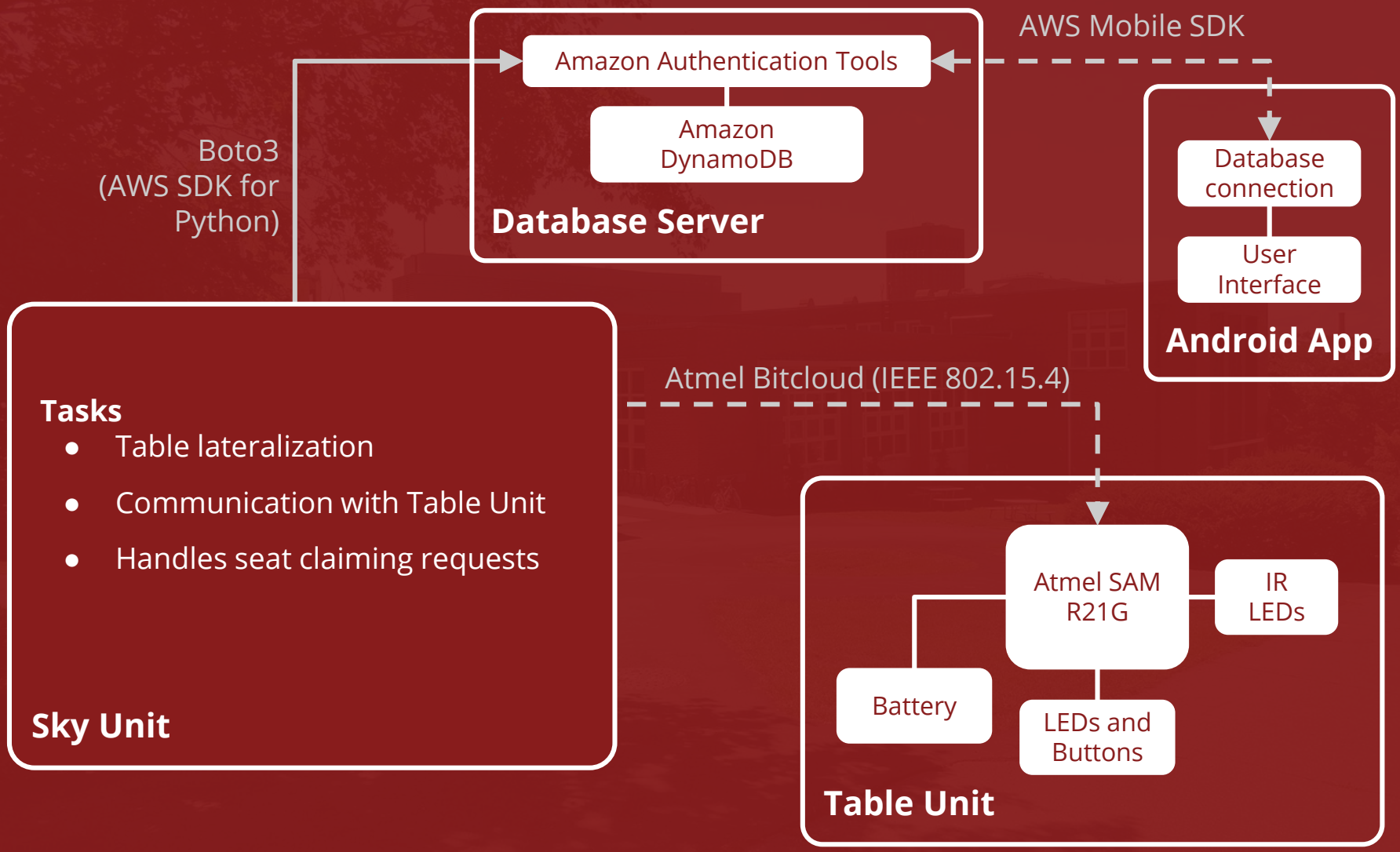
Redesigned Solution: Block Diagram



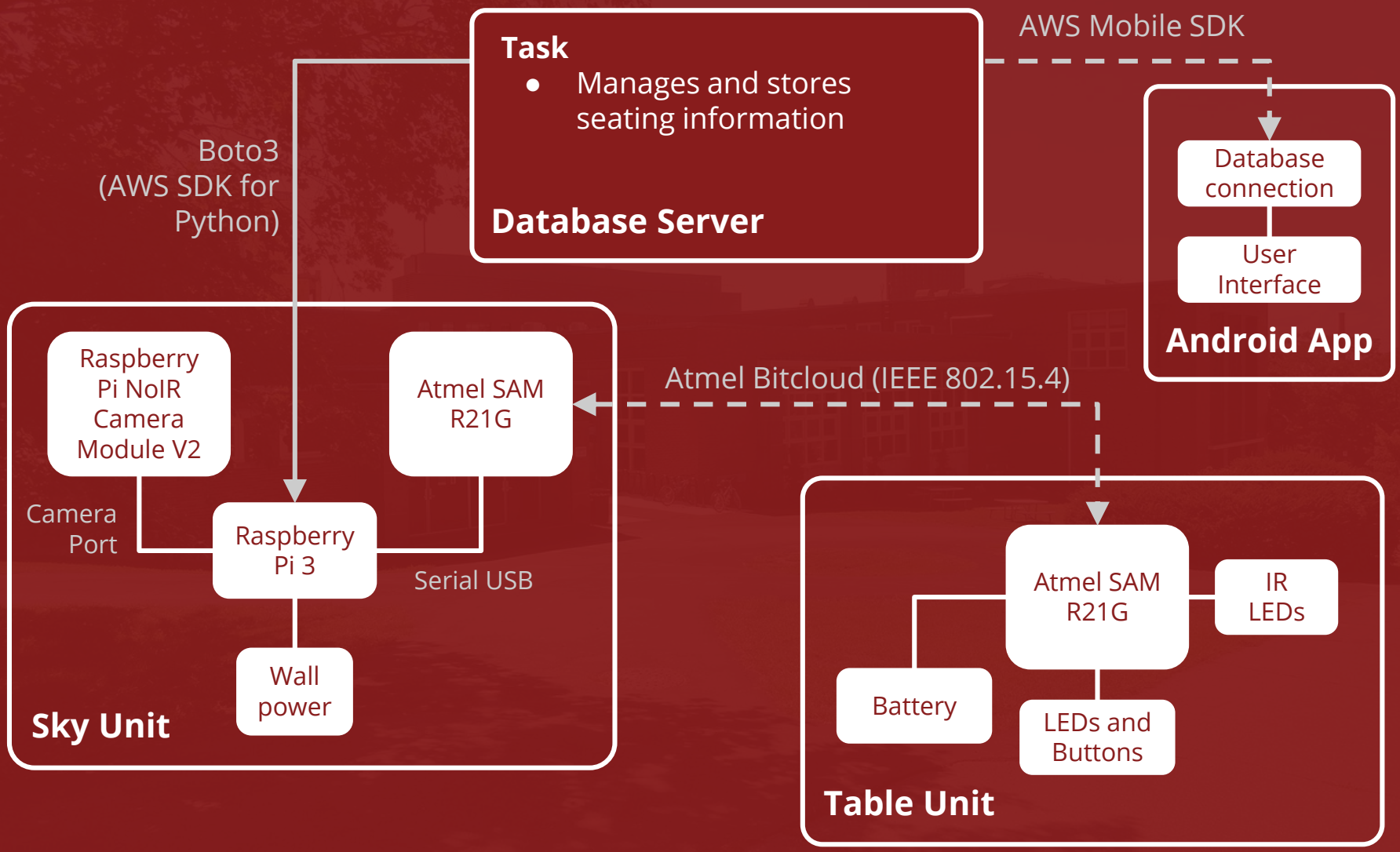
Redesigned Solution: Block Diagram



Redesigned Solution: Block Diagram



Redesigned Solution: Block Diagram



Redesigned Solution: Block Diagram

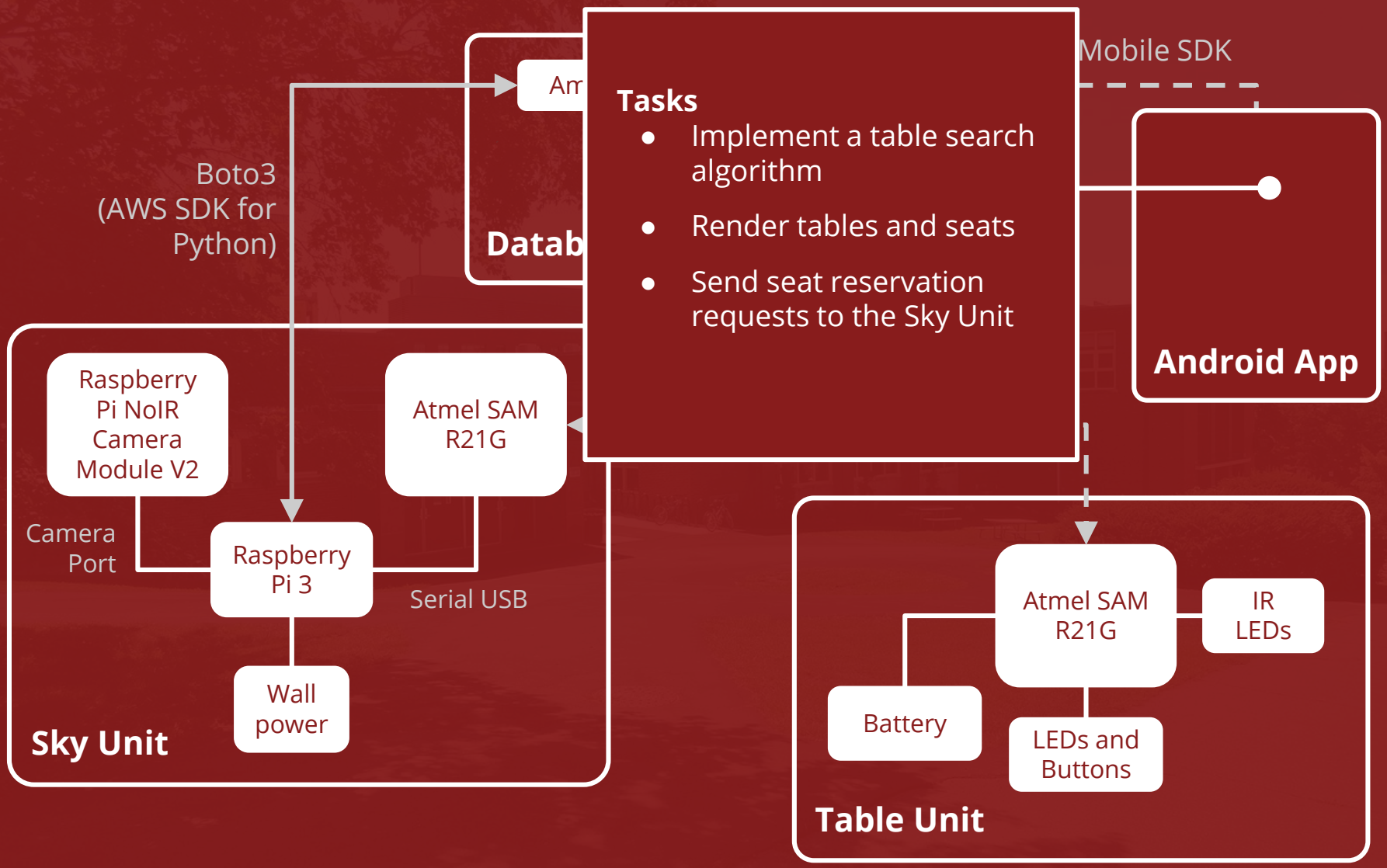


Table Unit: Choosing the MCU

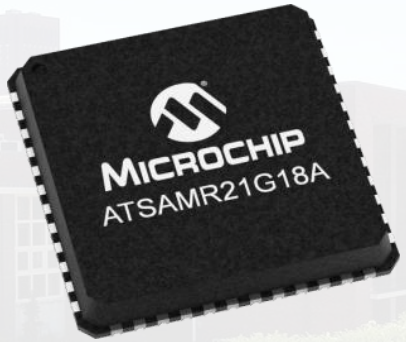
AIM: Low powered, wireless communication between many devices & cheap.

Microcontroller	Cost	Pros	Cons
Atmel SAM R21G	$5.2	2.4 GHz on-chip transceiver using IEEE 802.15.4 protocol Low powered	Relatively complex functionality
STMicroelectronics SPWF04	$15.8	High data rate with SoC using IEEE 802.11/b/g/ protocol Powerful functionality	Relatively power hungry Expensive for application High Complexity
Texas Instruments CC430 w/ MSP430	$15	Simplistic Communication Low Powered	Expensive for application Low support with stacks and protocol
ON-Semiconductor AX8052F143-3-TB05	$4	Low Powered Cheapest device available	Extremely low data-rate and range Low support with stacks and protocol

Table Unit: Atmel SMART SAM R21

Functions of Table Unit

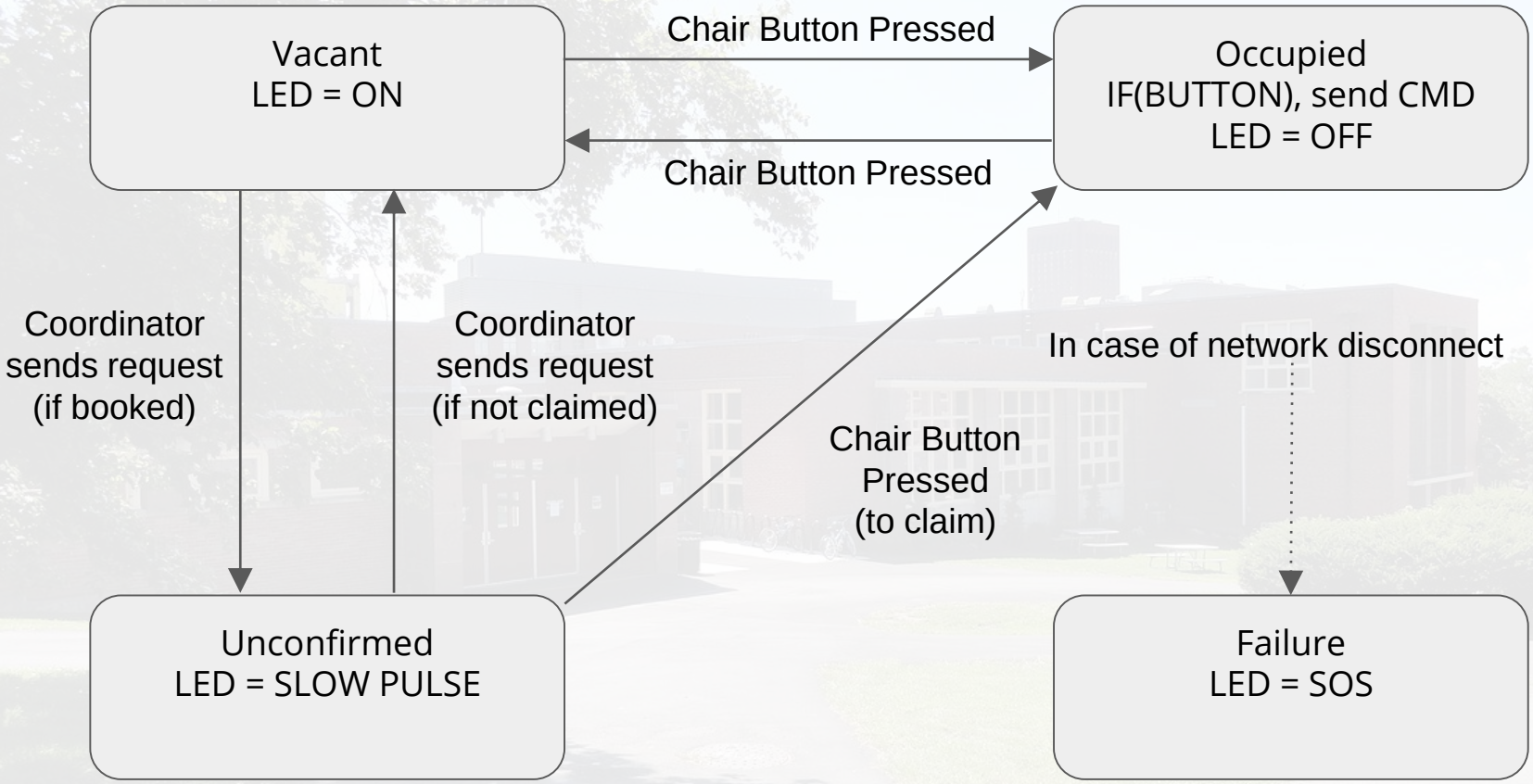
- Connect to Coordinator Node at Sky Unit in a network
- Change state of LED based on user input.
- Send the changed state to Coordinator Node.
- Receive data from Coordinator Node and process it.



Functions of Coordinator

- Network and data management of the Table Units.

Table Unit: State Diagram



Table/Sky Unit Communication Protocol

Table unit and Sky unit must communicate in order for system to maintain current, working knowledge of seating status

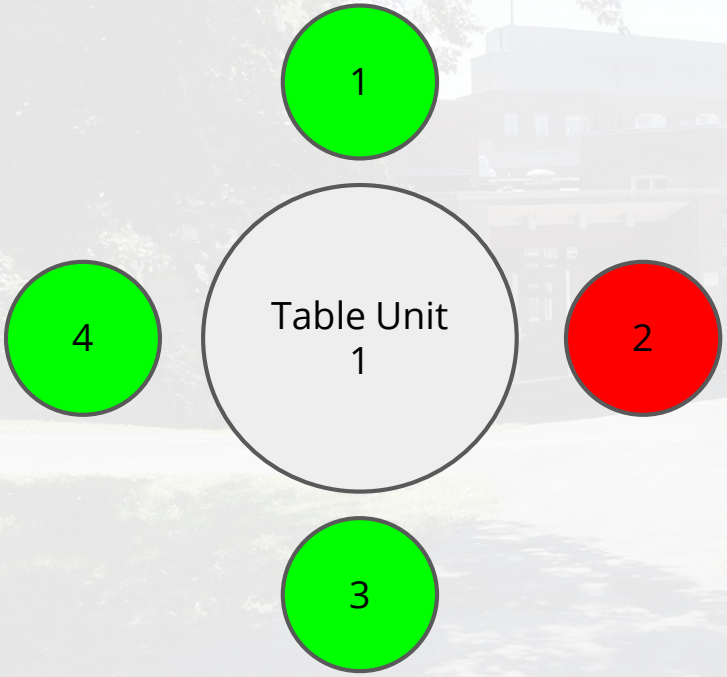
- Table unit must send data concerning:
  - Table ID
    - table unit address
    - seat number
  - State of seat (occupied, vacant)

- Sky unit must send data concerning:
  - Table ID
    - table unit address
    - seat number
  - Command type
    - new seat state, if change is needed

Table/Sky Unit Communication Protocol

Example

Sky unit communicating to table unit after a seat is reserved at Table 1



Payload:

01 122

addr
pin

cmd

data

Database Server

- Started with free mySQL database hosted on Amazon RDS
  - Faced Difficulties with...
    - Secure Pi-database connection
    - Table flexibility
    - App-database connection
- Moved on to DynamoDB
  - In-house, fully integrated Amazon system
  - Manages Users, Roles, and Rules for Security
  - Support for real-time, automatic scaling
  - Well documented and supported

Server	Pros	Cons
mySQL on Amazon RDS	Simple, Free, and partly integrated with AWS	Harder to access, less scalable, no support
Amazon DynamoDB	Powerful, flexible, and fully integrated with AWS	Not Free

Phone App: Rendered Map

- "View" frames scale to size of device
- Loaded tables scale to available canvas without distortion
- App "draws" (fetches, filters, and renders) data in under two seconds
- App draws on startup, then refreshes…
  - On party size change
  - On user action (swipe down)
  - Every 30 seconds
- Connection to Database is authenticated by AWS
- Pan, Zoom, and other usability features not yet implemented

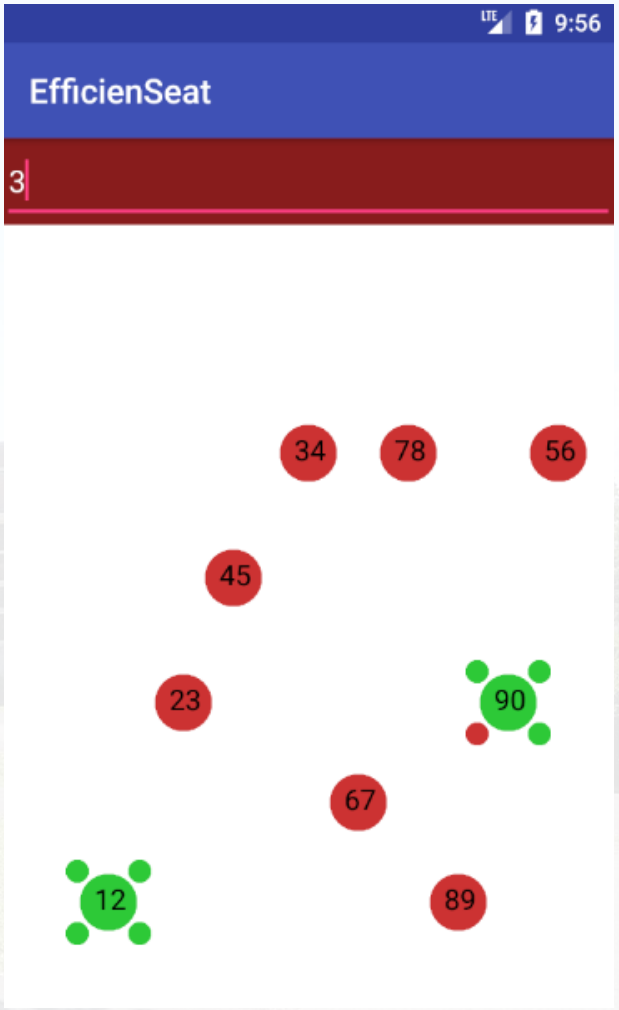


Table Localization: Concept

Table Unit position found by:

1. mapping the boundaries of the room
2. Generating a grid
3. Finding table unit on grid

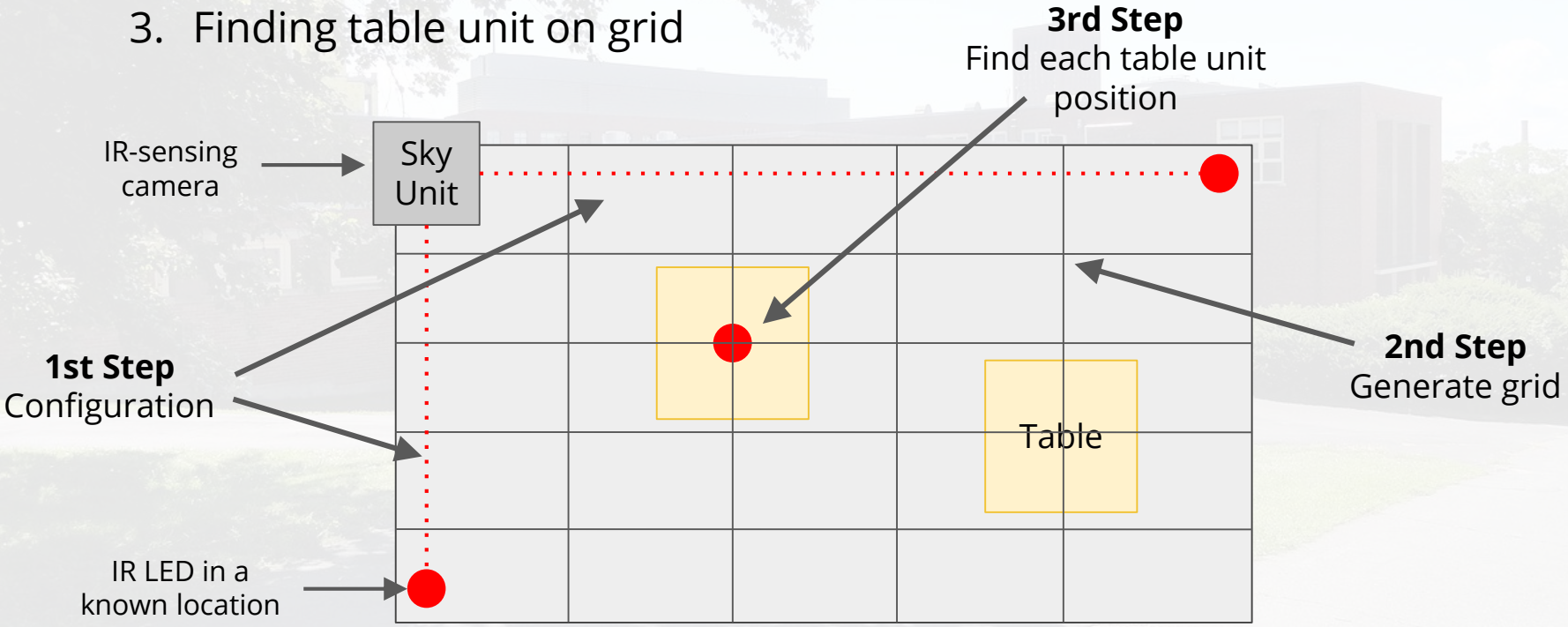
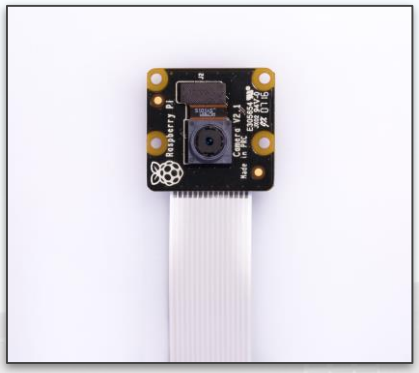


Table Localization: Components

Raspberry Pi NoIR Camera V2

- 8 megapixel resolution
- Blue filter (filter out red and green light)
- Synonymous with a SLR 35mm lens (considered a wide angle lense)

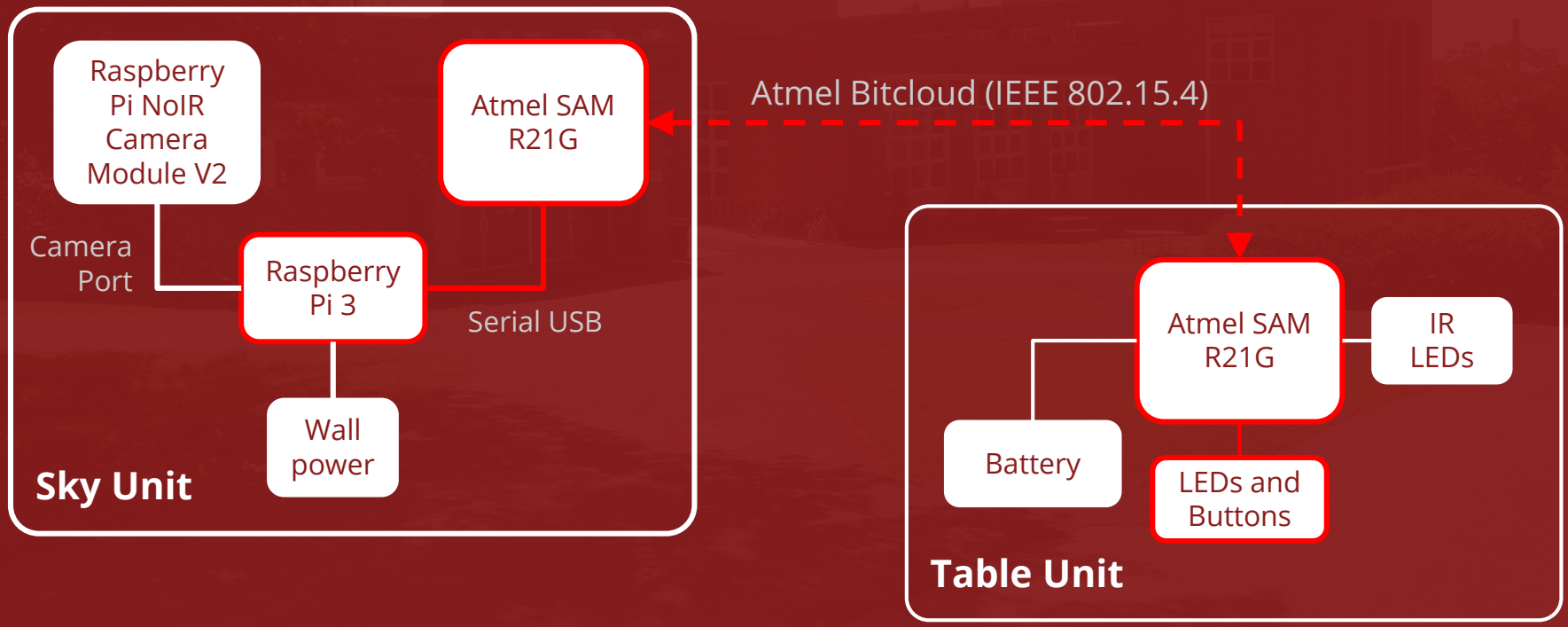


IR LEDs - 10mm

- 30-degree beamwidth
- Power consumption: 45mW
- Infrared Ray: 850nm (Near-Infrared Range)

Demo 1: Table Unit Functionality

- ➔ Push a button on the table unit, LED turns OFF or ON, data gets transmitted and displayed on terminal
- ➔ Type in a command on the computer, see response on table unit



Table/Sky Unit Communication Protocol

Payload

2 bytes	1 byte	1 byte	1 byte
addr	**cmd**	**data**	**pin**

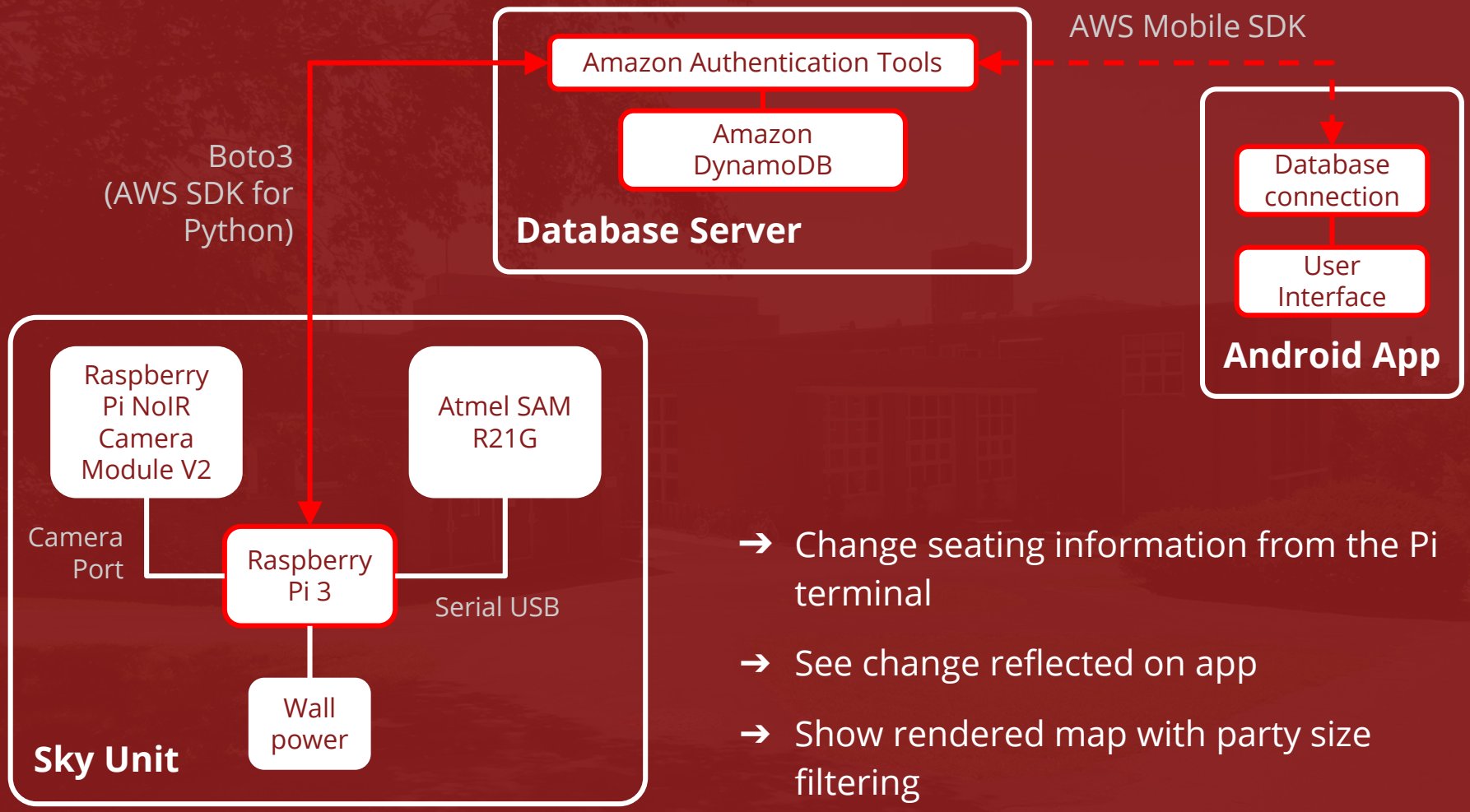
Definitions

ID	**Command**
0	Invalid
1	Change table state
2	Sleep
3	Wake up
4	Turn IR LED on

ID	**Data Entry**
0	Vacant
1	Occupied
2	Reserved
`3	About to expire
4	Sleep
5	Turn IR LED on

ID	**Pin Name**
0	Invalid
1	Seat 1
2	Seat 2
3	Seat 3
4	Seat 4
5	All Seats

Demo 2: App and Server Functionality



- ➔ Change seating information from the Pi terminal
- ➔ See change reflected on app
- ➔ Show rendered map with party size filtering

CDR Deliverables

- Demonstrate full system communication
  - Hitting the button on the table unit is reflected on the app and vice versa
  - Raspberry Pi as hub for seat claiming requests
- Prototype of table unit
  - Actual hardware for buttons and LEDs, battery
- Demonstrate table localization ability
  - Create a grid for a rectangular room
  - Determine position of a table unit

Looking Forward