Assignment 2

Wishing Well

Due Date: Sunday, March 22, 2020 @ 11:55pm

ECE 4564 - Network Application Design

Learning Objectives

Message Broker

- AMQP Protocol
- RabbitMQ
- Direct Exchange

Callback Routines

JSON/BSON Data

Data Persistence

- noSQL Database

Bluetooth

Raspberry Pi GPIO

Place-Specific Computing

- Place-specific computing is the study of the use of mobile information and communication technologies as they relate to context in terms of place.

- It is the design of interactive digital systems and services for specific places.

- A place-centric perspective for the design of digital systems and services, in which functionality, as well as information content, emanate from the place of use

“Place-Specific Computing:

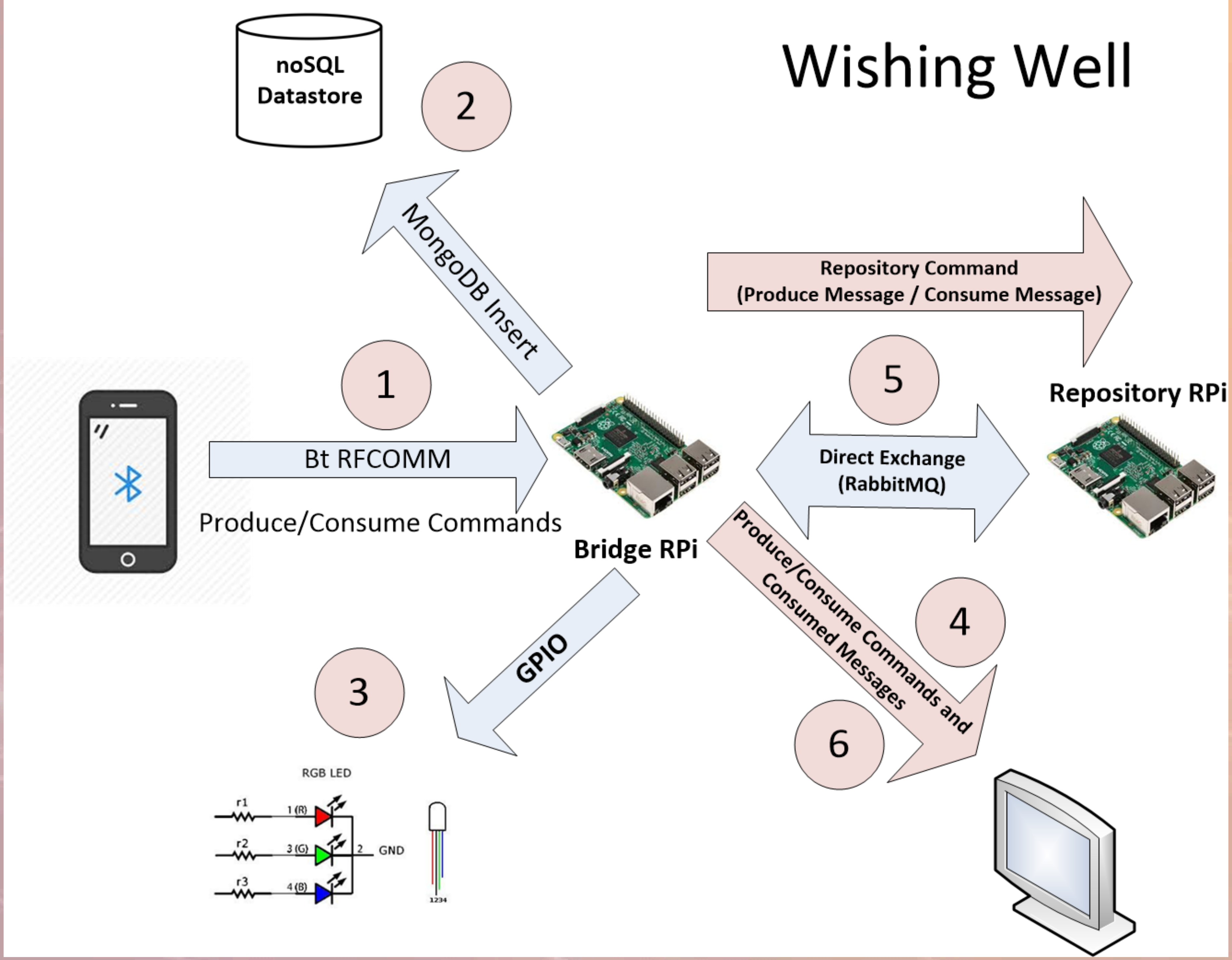
A Place-centric Perspective for Digital Designs”

John Messeter

Wishing Well

- Your system is replicated across campus
- Each instance is associated with a physical space
  - Examples:
    - Squires
    - Lane Stadium
    - Torgersen Hall
- System supports
  - Deposit (produce) messages about the associated physical space
  - Retrieval (consume) messages deposited by others,

System Overview



Wishing Well

noSQL Datastore

2

MongoDB Insert

1

Bt RFCOMM

Produce/Consume Commands

Bridge RPi

Repository Command
(Produce Message / Consume Message)

5

Direct Exchange
(RabbitMQ)

Repository RPi

Produce/Consume Commands and Consumed Messages

4

6

GPIO

3

Assignment Overview

- Mobile Phone
  - Sends a command to place a message in or remove message(s) from RabbitMQ message queues on the Repository Rpi
  - Uses terminal application over Bluetooth (RFComm)

- Bridge RPi
  - Receives message instruction from mobile phone
  - Issues a produce request to Repository Rpi –or-
  - Issues a consume request to Repository RPi
  - Maintains messages in a persistent (MongoDB) datastore
  - Indicates command status using RGB LED
  - Displays commands and consumed messages on monitor

Assignment Overview

- Repository Rpi (server)
  - Runs the RabbitMQ service
  - Runs the RabbitMQ management plugin

MessageCommands

```
p:exchange+queue message
c:exchange+queue

p:place+subject message
c:place+subject
```

p = produce c = consume

Produce a message:

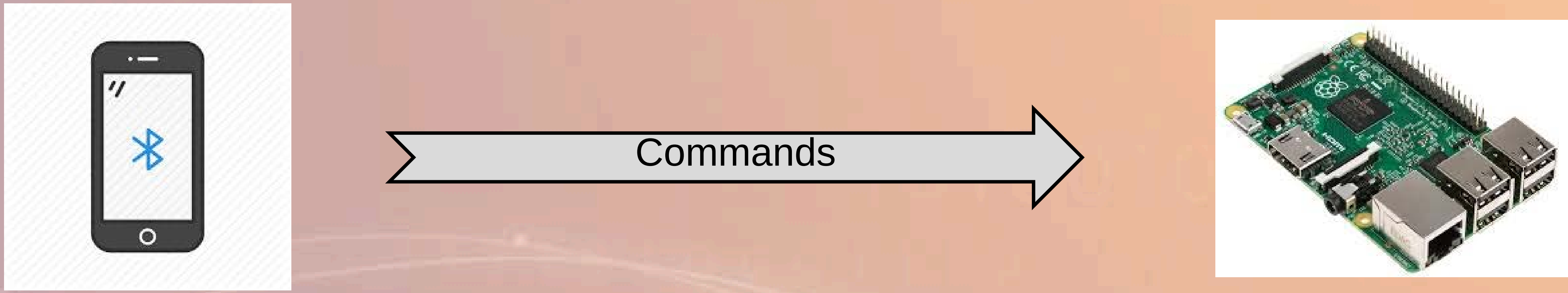
p:Squires+Rooms “I like the comfortable chairs on 3rd floor”

p:Library+Wishes “I wish the lines were shorter here”

Consume messages:

c:Goodwin+Classrooms

Mobile Phone



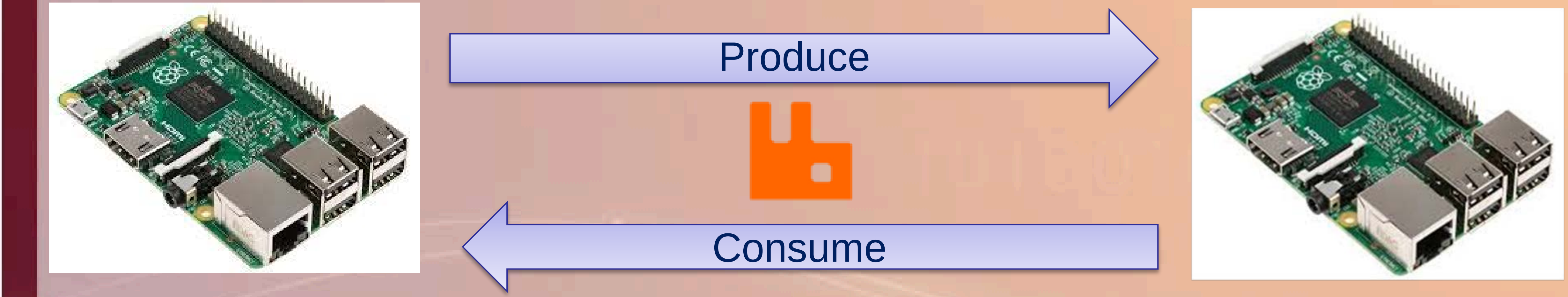
Commands

- The mobile phone sends produce/consume commands to the Bridge RPi via Bluetooth terminal session
- Message client operations
  - Send Produce (p) command to the Repository Rpi via Bridge RPi
  - Send Consume (c) command to the Repository Rpi via the Bridge RPi

Bridge RPi

Bridge RPi

Repository RPi



- Filename : bridge.py
- Receive message command from mobile phone
- Place message command in noSQL (MongoDB) datastore
- Indicates command status using RGB LED
- Displays commands and consumed messages on monitor
- Sends message instruction to the Repository Rpi via a direct exchange
- Receives replies from the Repository RPi via a direct exchange

Bridge RPi Initialization

```
python3 bridge.py -s <Repository_RPi_IP>
```

Example:

```
python3 bridge.py -s 192.168.1.128
```

MongoDB Format

```
Place : Squires
Command : p:Squires+Rooms “I like the comfortable chairs on 3rd floor”

Warehouse : Squires
Collection : Rooms
                            MongoDB Document
{
    "Action": "p",
    “Place”: “Squires”
    "MsgID": "02$1476123693.1855621",
    "Subject": “Rooms",
    "Message": "I like the comfortable chairs on 3rd floor"
}

MsgID : "team_#" + "$" + ticks
                  ticks = time.time()
```

GPIO - LED

Indicates system status:

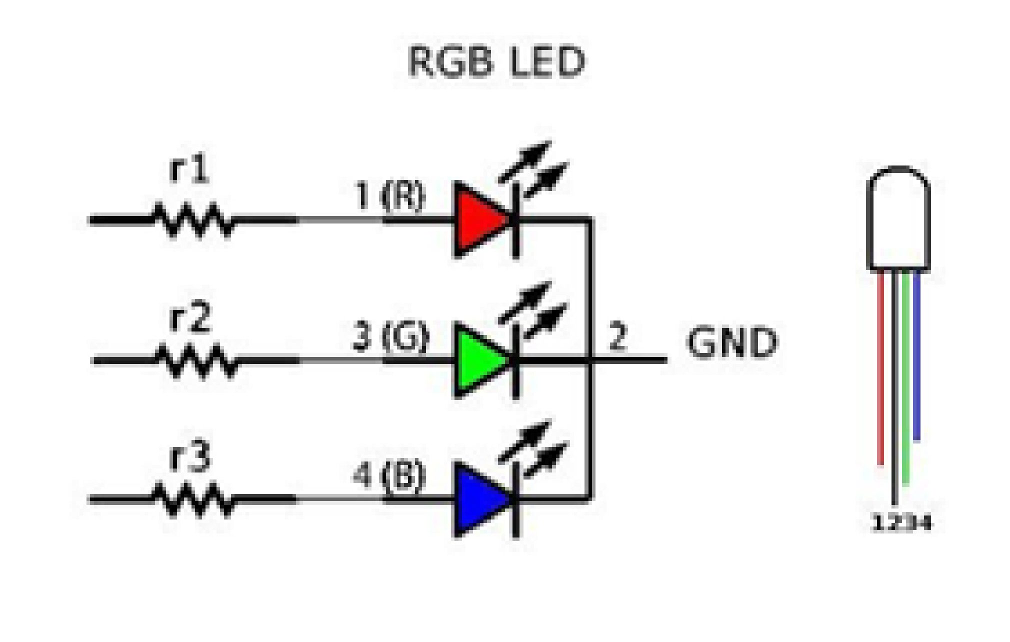
White – Waiting for a command

Red – Received publish request

Green – Received consume request

Blue – MongoDB store operation

Note: use timing delays



Repository RPi

- Manages RabbitMQ messages via direct exchange and queues

RabbitMQ Install

Installation

sudo apt-get install rabbitmq-server

Start management console

sudo rabbitmq-plugins enable rabbitmq_management

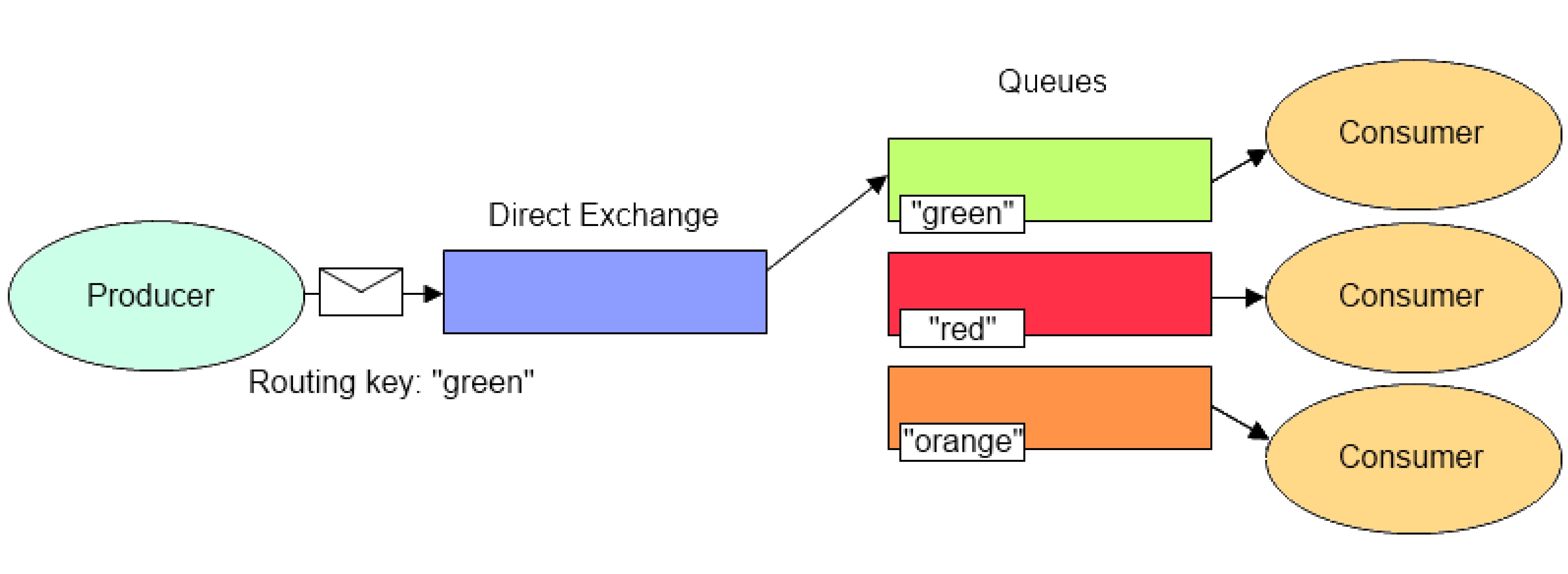
Config administrator

sudo rabbitmqctl add_user [newuser] [password]

sudo rabbitmqctl set_user_tags [newuser] administrator

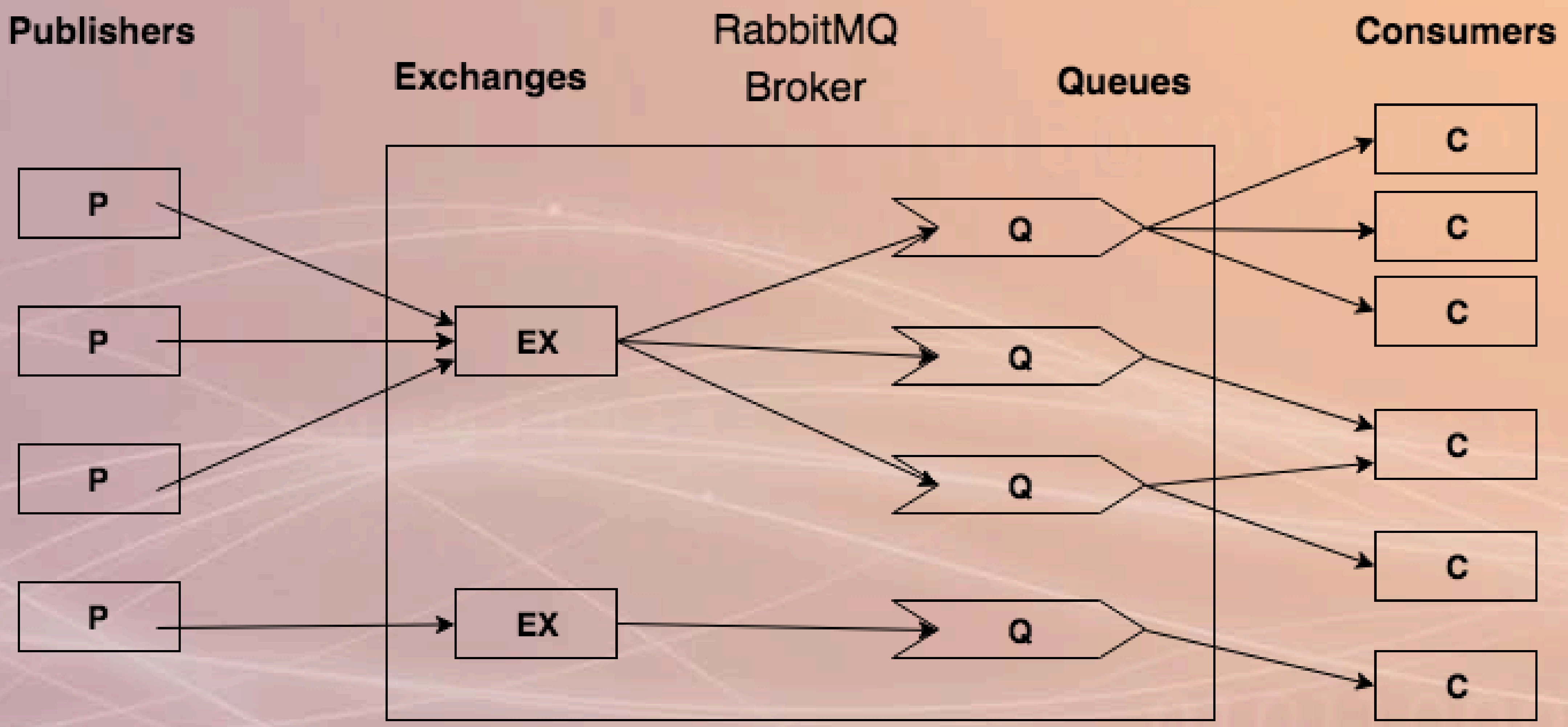
sudo rabbitmqctl set_permissions -p / [newuser] ".*" ".*" ".*"

RabbitMQ Direct Exchange



Use Direct Exchange for this assignment

System Overview



Exchanges and Queues

- Exchange
  - Queue
  - Queue
- Place
  - Subject
  - Subject

Supported by your system

- Squires
  - Food
  - Meetings
  - Rooms
- Goodwin
  - Classrooms
  - Auditorium
- Library
  - Noise
  - Seating
  - Wishes

RabbitMQ

```
Produce Example:
    Place: Library
    Subject: Wishes
    Message: “I wish I remembered their name”


channel.exchange_declare(exchange=Place,
                         exchange_type='direct')

channel.basic_publish(exchange=Place,
                      routing_key=Subject,
                      body=Message)
```

RabbitMQ

```
Consume Example:
    Place: Squires
    Subject: Food

channel.queue_bind(exchange=‘Place',
                       queue=queue_name,
                       routing_key=Subject)

def callback(ch, method, properties, body):
    print("%r:%r" % (method.routing_key, body))

channel.basic_consume(callback,
                      queue=queue_name,
                      no_ack=True)

channel.start_consuming()
```

Repository RPi Initialization

- RabbitMQ instance starts at boot-up
- RabbitMQ management plugin starts at boot-up
- Exchanges and queues pre-configured using management plugin

Grading

GTA will provide grading rubric

Python Style

Follow style guide PEP0008 when writing and commenting your code

https://www.python.org/dev/peps/pep-0008/

What You Turn In

All assignments must be submitted through Canvas, no later than the due date of Sunday, March 22, 2020 @ 11:55pm

Note: Teams will receive a 10 point deduction per day past the due date

Your assignment should be a single tar gz (tgz extension) which contains the following:

- All source code you wrote for this assignment
  - Python code running on Bridge Rpi
    - Client code identified as “bridge.py”
- Report (PDF file)

Be sure to name the tar gz file as follows: HW2_TeamXX.tgz

Assignment References

RabbitMQ Tutorials

- RabbitMQ – Direct Exchange Tutorial
- RabbitMQ Management Plugin
- Pika

Note: Install RabbitMQ and pika versions as discussed in class

MongoDB

- Follow installation instructions discussed in class
- JSON and BSON

Mobile Phone – Bt terminal application

Bluetooth on the Raspberry Pi

Android – BlueTerm

iOS – Bluetooth Terminal

Academic Integrity

- For this assignment, it is expected that a team’s work is their own.
- The code you turn in must be your own (i.e. you need to have written your assignment).
- You are allowed to copy and paste example code from other websites, but you must include a comment in your code that attributes the website you copied the code from (i.e. original author’s name and URL to the original code).
- You can discuss the assignment with other teams.
- However, you cannot just tell another team the answer to a particular problem.

Final Thoughts

In many cases, engineers are expected to just make things work given a particular design constraint (e.g. software package to use or are limited to a particular hardware platform).

You will likely run into similar situations in this class while designing and implementing your assignments and project.

.

When you're stuck, try searching online for a solution. Many times others have tried something similar and documented their experiences for others to learn and benefit from

Do not publically post answers to assignments, or example code until after the assignment due date.

Contact your instructor or GTA as soon as you encounter a problem you're unable to solve.

Don't wait until right before the assignment is due.