SE-IOT: Internet of Things

User Interfaces

Derek Kiong
dkiong@nus.edu.sg

User Interface

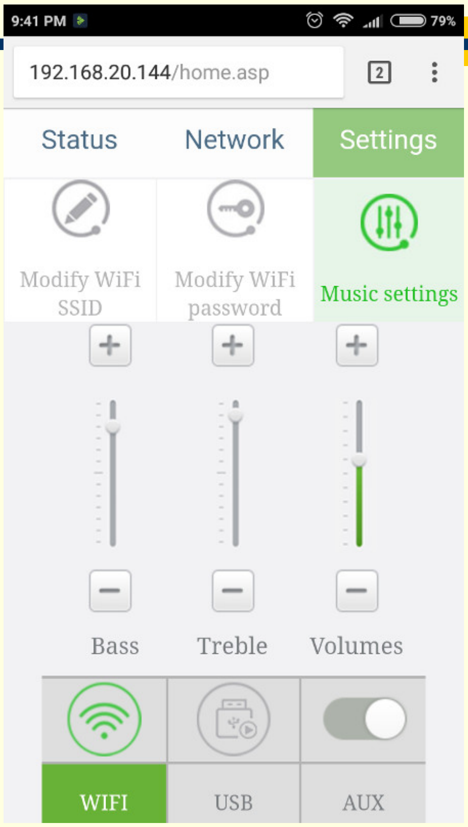
- Engineering compromise
  - Simplify device usage
  - (includes configuration and setting options)
  - Minimise (construction) cost
    - Component cost
    - Development cost

DLNA speaker

- How to setup WiFi without display panel?

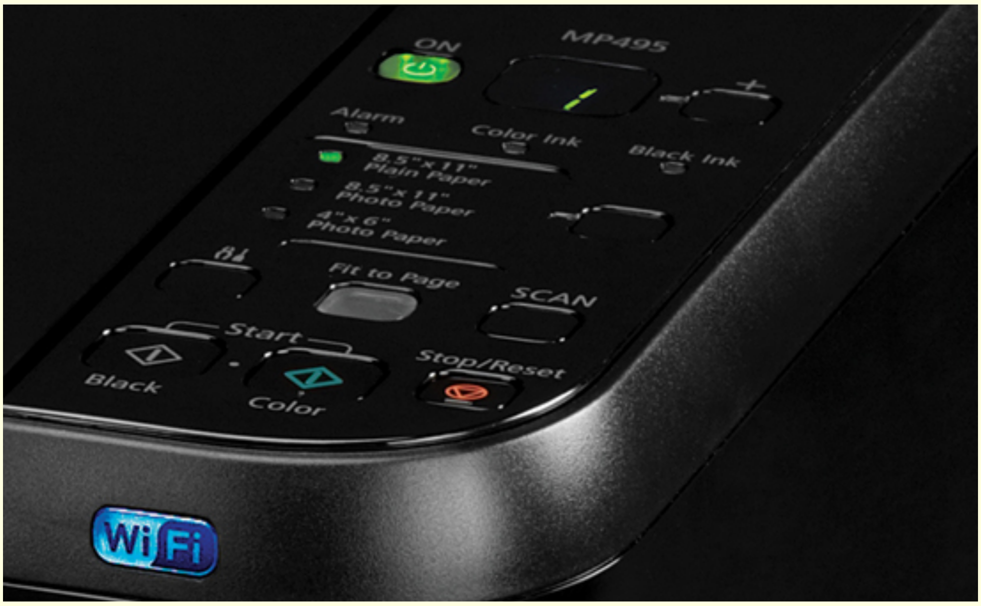
DLNA speaker

- Setup via web browser or mobile app –
- (WiFi setup involves hotspot mode before access point mode.)



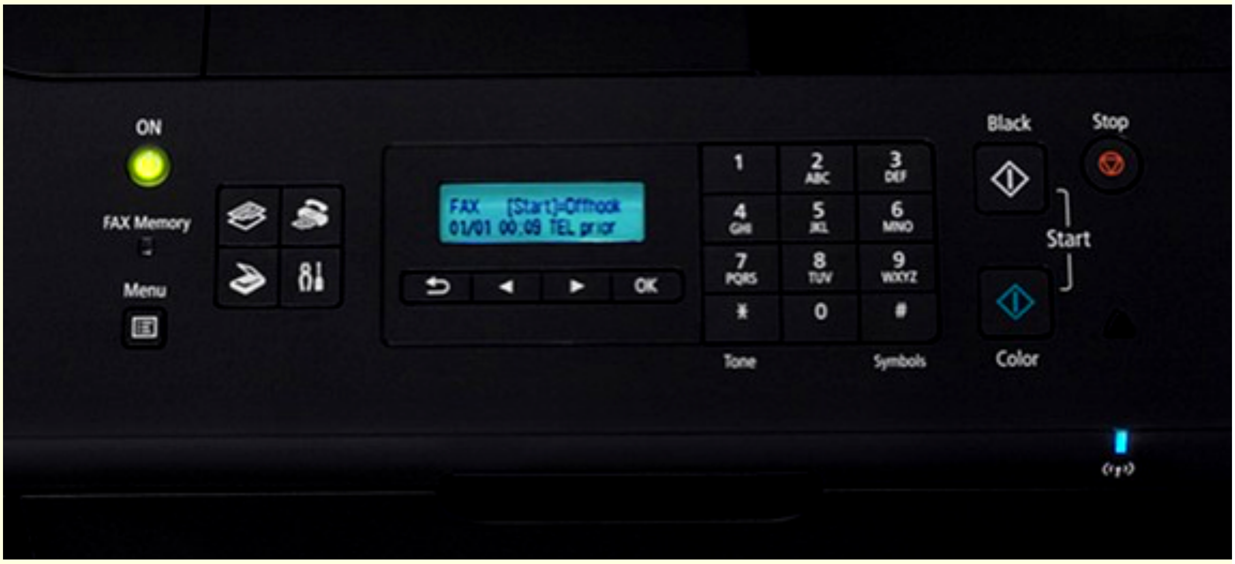
WiFi-enabled printer

- No substantial output feedback except LEDs and single segment display

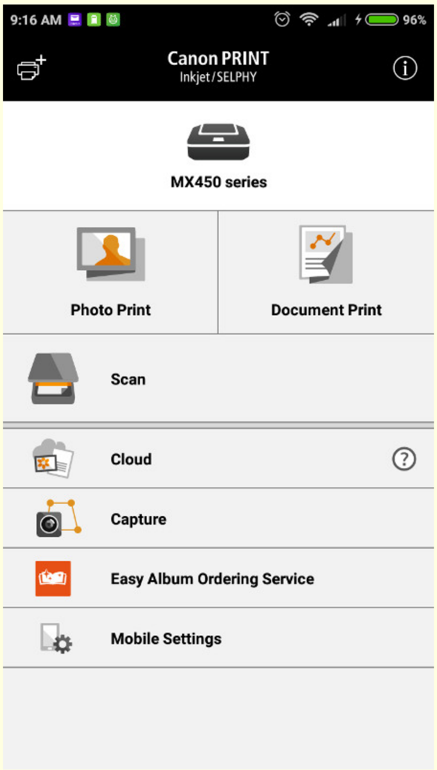


WiFi-enabled printer

- More substantial output feedback with LCD display



More functionality with (remote) screens



9:16 AM 96%

Canon PRINT
Inkjet/SELPHY

MX450 series

Photo Print

Document Print

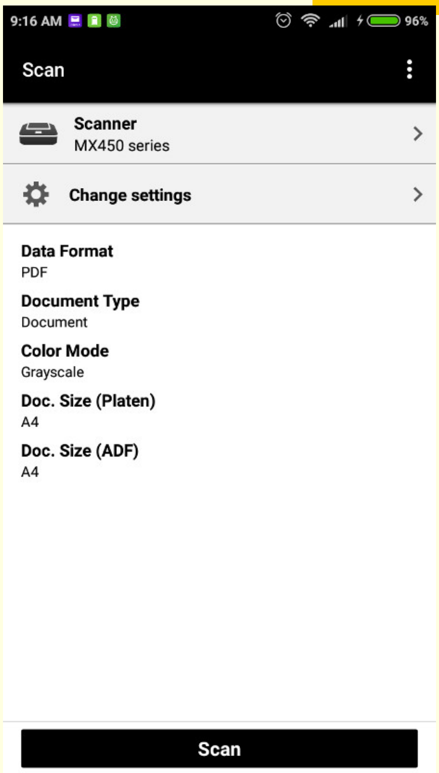
Scan

Cloud

Capture

Easy Album Ordering Service

Mobile Settings



9:16 AM 96%

Scan

Scanner
MX450 series

Change settings

Data Format
PDF

Document Type
Document

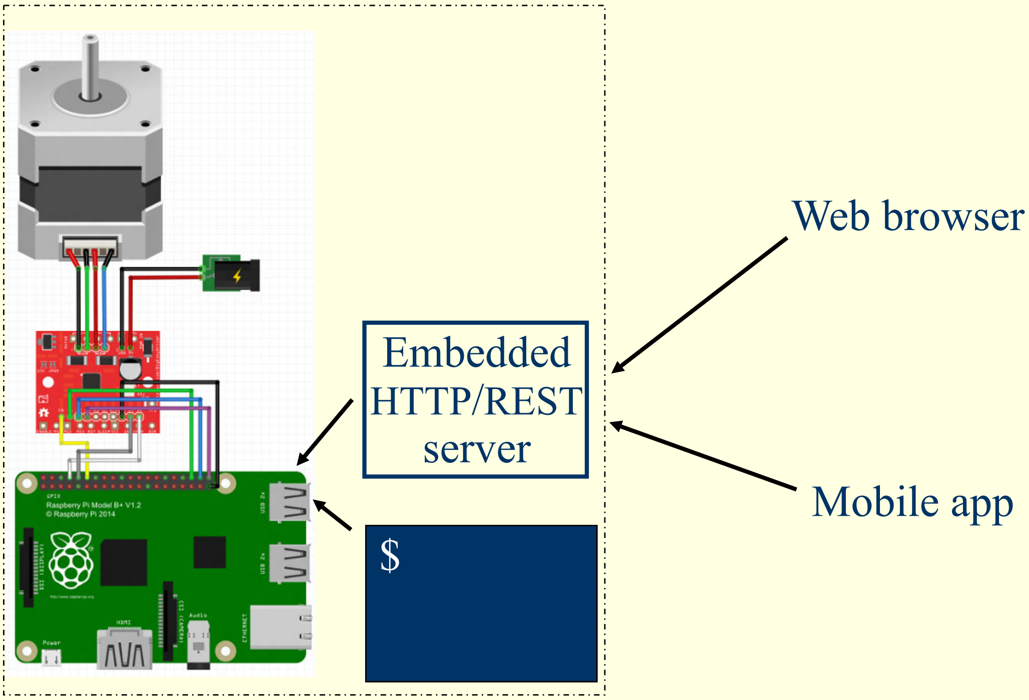
Color Mode
Grayscale

Doc. Size (Platen)
A4

Doc. Size (ADF)
A4

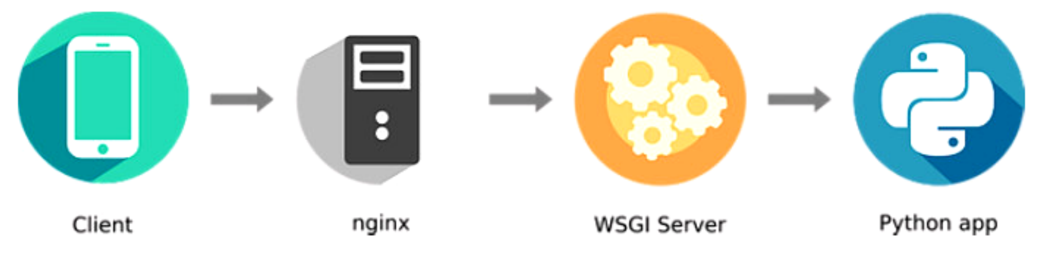
Scan

Prototype vs usable product



HTTPD server

- **`nginx`** provides small footprint http server
- **`php5-fpm`** provides PHP engine for nginx
- **`nginx`** hosts Django apps via uWSGI module



HTML Form

```
<h1>Relay Test</h1>
<p>
<form action="relay.php" method="get">
<hr>
Lamp
<input type="radio" name="lamp" value="0" checked >0
<input type="radio" name="lamp" value="1" >1
<input type="radio" name="lamp" value="2" >2
<input type="radio" name="lamp" value="3" >3
<hr>
State
<input type="radio" name="state" value="0" checked >Off
<input type="radio" name="state" value="1" >On
<hr>
<input type="submit" value="Submit">
</form>
```

PHP script

```
<body>
<?php
$output = shell_exec
("python /home/pi/src/Relay.py {$_GET['lamp']} {$_GET['state']}");
echo "<pre>$output</pre>";
?>
</body>
```

Django Framework

- Django is a Python web application framework
  - Django URL dispatcher

```
urlpatterns = patterns('',
    url(r'^$', 'mysite.views.home'),
    url(r'^product$', 'mysite.views.listproducts'),
    url(r'^product/(.+)$', 'mysite.views.product'),
)
```

  - Function
  - Template
  - also includes database integration and user roles and authentication

(Python) Function

```
def product(request, code):
    connection = None
    data = dict()
    data['title'] = 'Product details: '+code
    try:
        cons = sqlite3.connect('/home/pi/src/mydatabase.db')
        with connection:
            cursor = cons.cursor()
            cursor.execute("SELECT * FROM Product WHERE Id=:Id",
                           {'Id':code})
            data['products'] = cursor.fetchall()
    except sqlite3.Error, e:
        print "Error %s:" % e.args[0]
        sys.exit(1)
    finally:
        if cons:
            cons.close()
    html = get_template('product.html').render(Context(data))
    return HttpResponse(html)
```

Django Template

```
<html>
<head>
<title>{{ title }}</title>
</head>
<body>
<h1>{{ title }}</h1>
<table border="solid">
{% for t in products %}
    <tr><td><a href="product/{{ t.0 }}">
            {{ t.0 }}</a></td>
        <td>{{ t.1 }}</td>
    </tr>
{% endfor %}
</table>
</body>
</html>
```

Audio and Voice

- Amazon Alexa Voice service
  `https://developer.amazon.com/alexa-voice-service`
- Recognise audio input
- Various skills to be invoked

Summary

- Embedded HTTP/HTTPS server allow for device setup/configuration or REST services
- Supports user authentication