

# PUNYA Framework: On the Fly Development & Deployment of Android Apps

---

Julius Adebayo  
CSAIL, MIT

# Mobile Ready

- **Smartphones** are becoming the primary computing platform for people's daily tasks
- Have been found to be useful especially for **disaster management and relief operations**



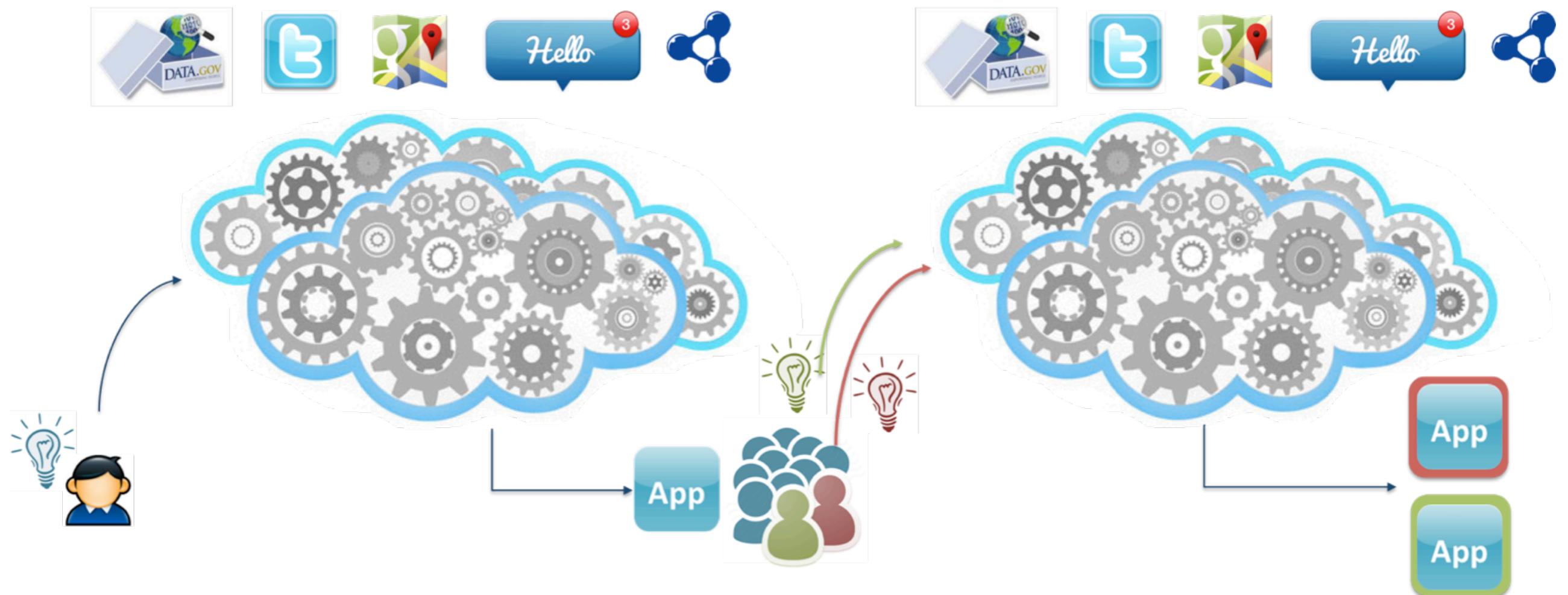
ref: <http://nhma.info/resources/android-apps/>

# Current Mobile Development Landscape

---

- ❖ Detailed requirements for the application
  - Difficult to prototype
- ❖ Budgeting for app development
  - \$75 ~ \$150 per hr for an experienced freelancer,  
8~10 weeks to design and complete an app
  - 24k ~ 50k (usd) for complete app
- ❖ Lengthy development time

# App Development Platform



# Critical Platform Requirements

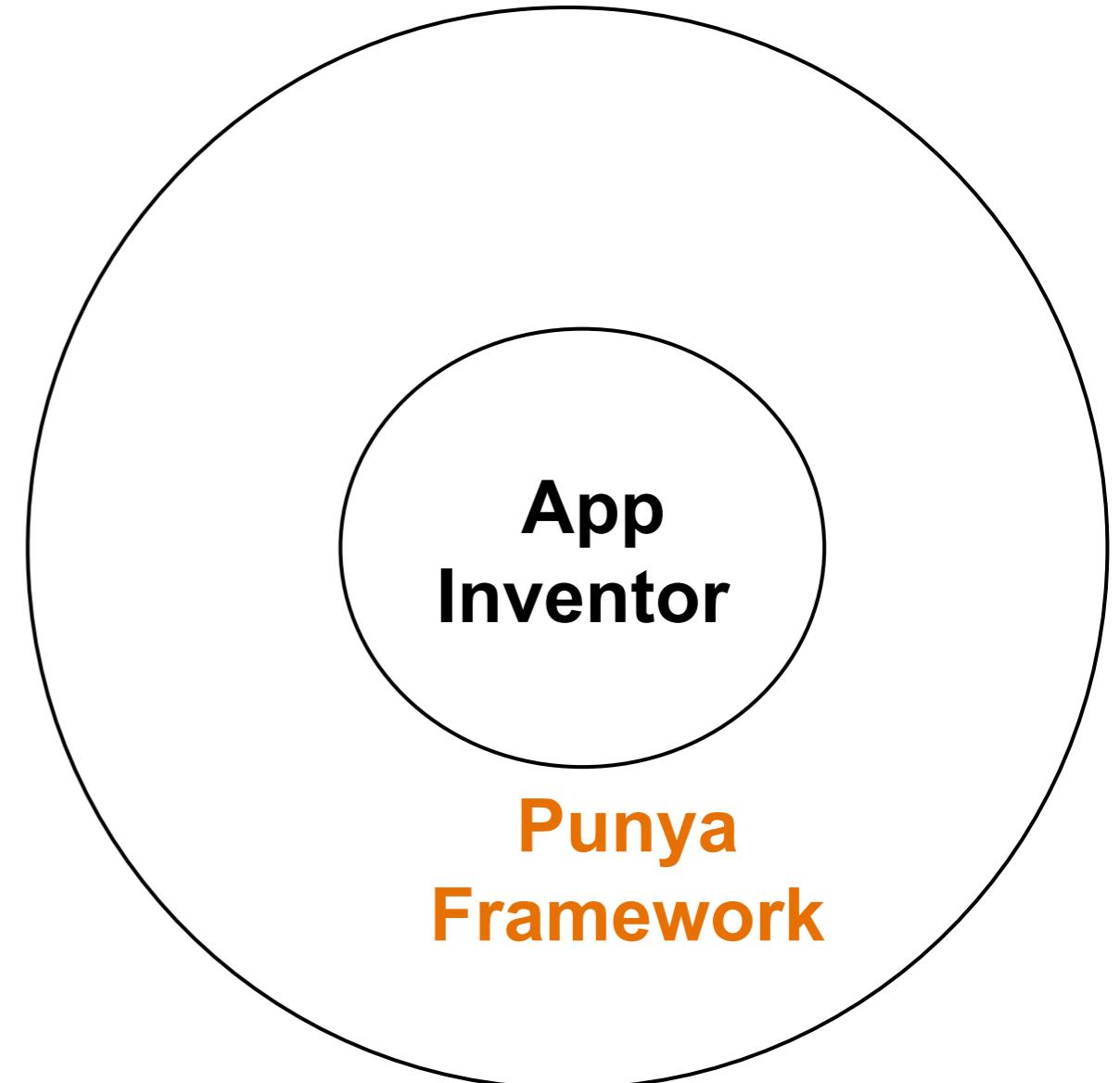
---

- **Requirement 1:** mobile apps are the best way to collect data, communicate, self-organize, etc. during a crisis
  - **Challenge:** Difficult to develop & deploy mobile apps **on the fly**
- **Requirement 2:** Apps need to process large amounts of (heterogeneous) data created & available
  - **Challenge:** Difficult to integrate (heterogeneous) data from different sources
- **Requirement 3:** Humanitarian focused apps require backend/server side processing
  - **Challenge:** Require significant setup time
- **Requirement 4:** Data analytics
  - **Challenge:** Data needs to be processed, analyzed, visualized to provide useful feedback to decision makers.

# Our Solution: Punya Framework

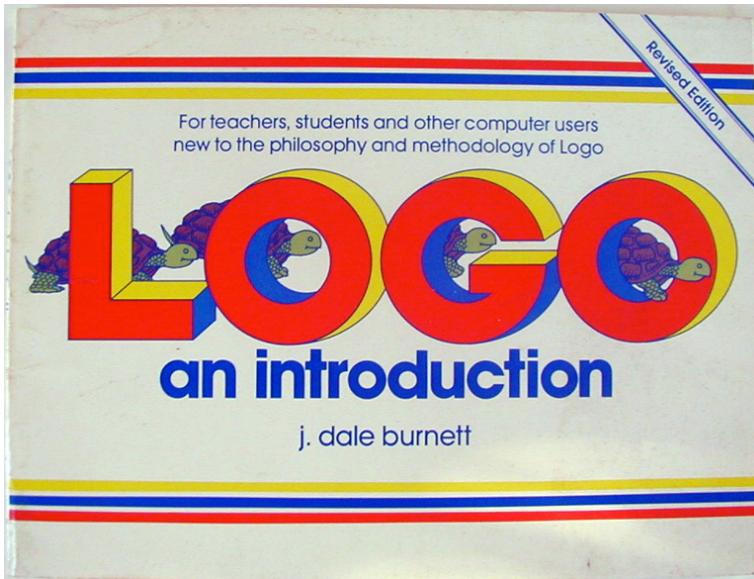
- ***App Inventor***

- Former Google Research Lab Education Project
- Block programming tool for non-programmer to learn about mobile programming
- Designed for students



# Our Solution: Punya Framework

## Block programming language



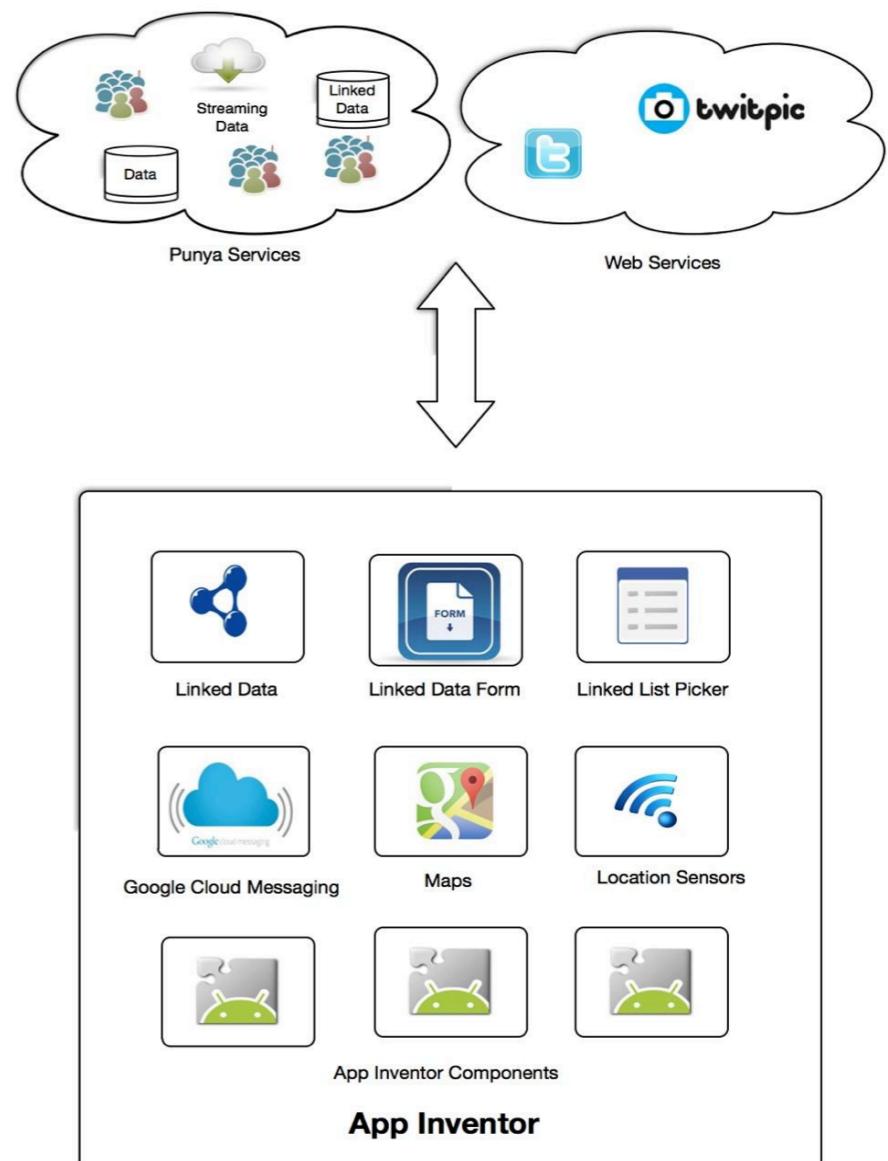
SCRATCH



MIT  
App Inventor

# Punya Framework

- Vertical integration to provide core functionality for **humanitarian contexts**
  - **Components**
    - Cloud messaging component
    - Linked Data components
    - Sensor components
    - Map component
  - **Services**
    - Cloud messaging services
    - Linked Data service
    - Stream data Query



**PUNYA Framework**

# Framework in Practice: Designer View

**HelloCat**

Screen1 ▾ Add Screen ... Remove Screen Designer Blocks

**Palette**

**User Interface**

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- TextBox
- TimePicker
- WebViewer

**Layout**

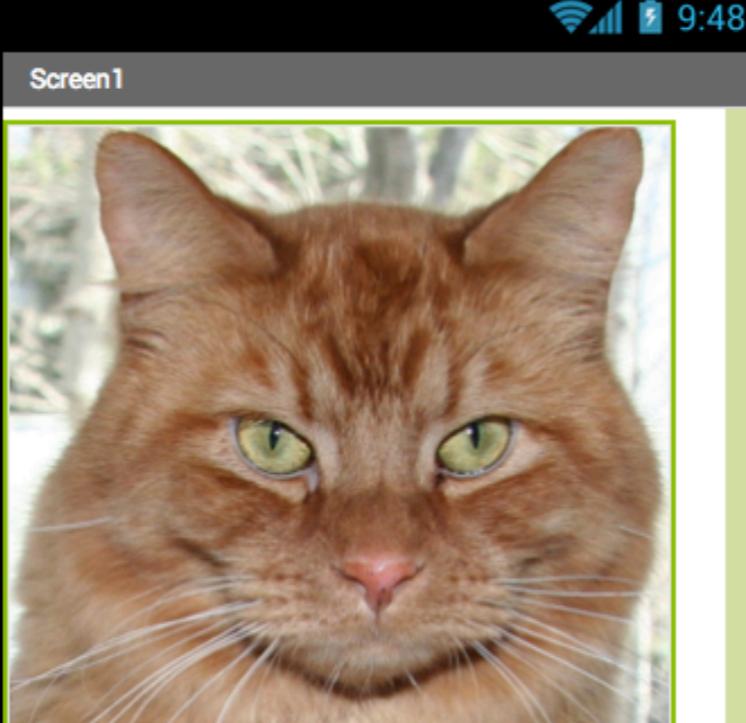
**Media**

**Drawing and Animation**

**Sensors**

**Viewer**

Display hidden components in Viewer



Screen1

**Components**

- Screen1
  - Button1
  - Sound1

**Properties**

**Button1**

- BackgroundColor: Default
- Enabled:
- FontBold:
- FontItalic:
- FontSize: 14.0
- FontTypeface: default
- Image: kitty.png...
- Shape: default
- ShowFeedback:
- Text:
- TextAlignment: center

**Non-visible components**

Sound1

# Framework in Practice: Blocks View

The image shows a Scratch-like interface for a programming framework. On the left, a 'Blocks' palette lists categories: Built-in (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), Screen1 (Button1, Sound1), and Any component. The 'Screen1' category and its items are highlighted with a red box. On the right, a 'Viewer' window displays six script blocks for a 'Button1' component:

- when **Button1**.Click
- when **Button1**.GotFocus
- when **Button1**.LongClick
- when **Button1**.LostFocus
- when **Button1**.TouchDown

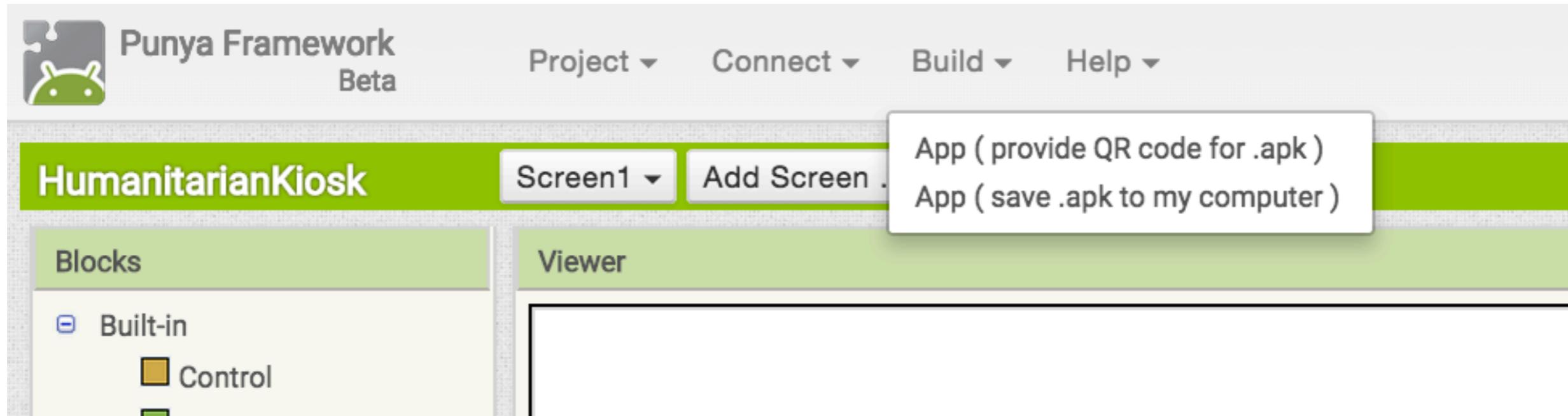
Each script block consists of a green 'when' hat block followed by a light blue 'do' slot.

# Framework in Practice: Logic

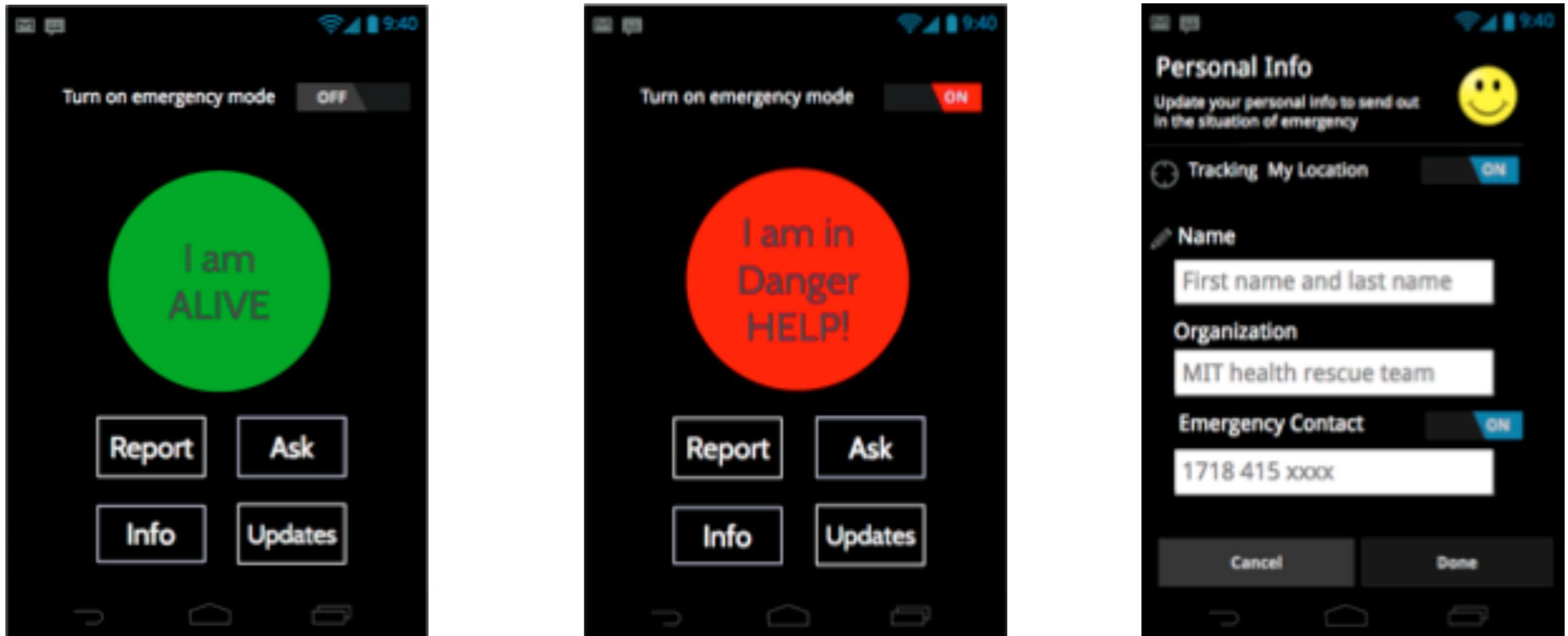
```
when Button1 .Click
do call Sound1 .Play
```

```
to getAllCountries
do set Web1 .Url to " http://api.rwlabs.org/v1/countries/ "
set Web1 .SaveResponse to false
call Web1 .PostText
text { "sort": ["name:asc"], "limit": "400" }
```

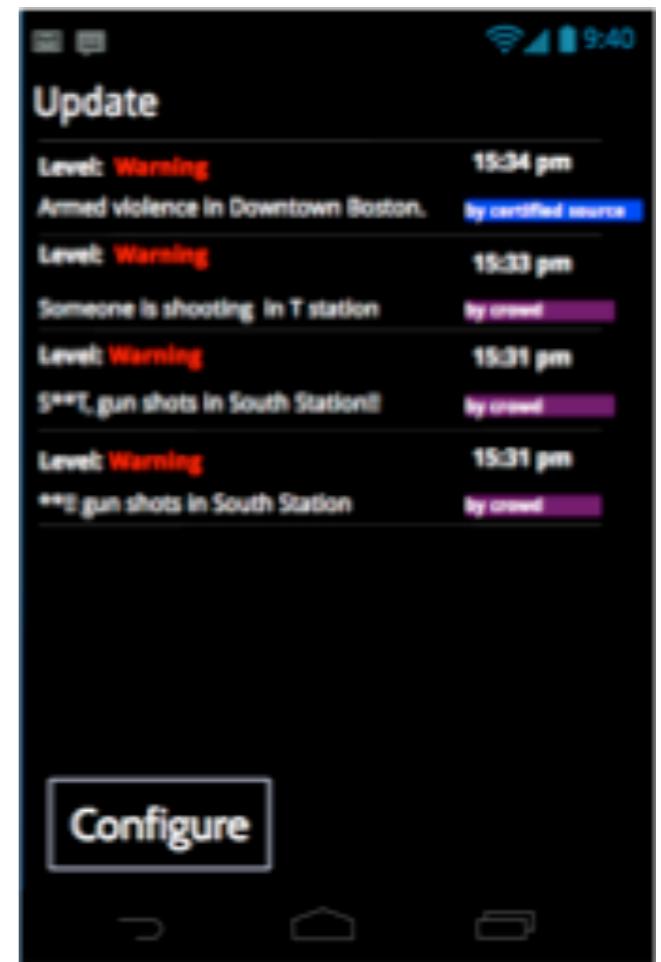
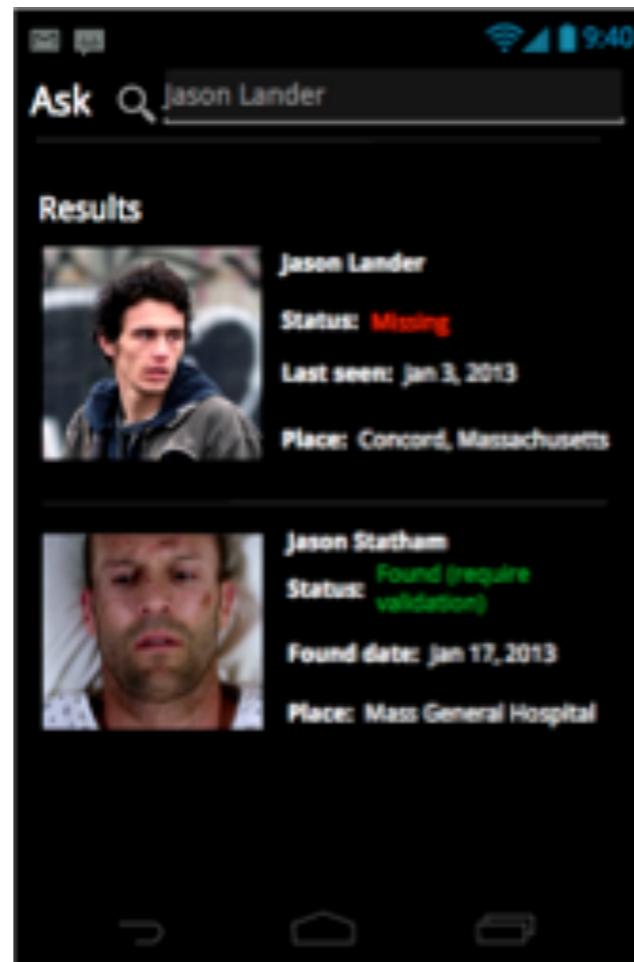
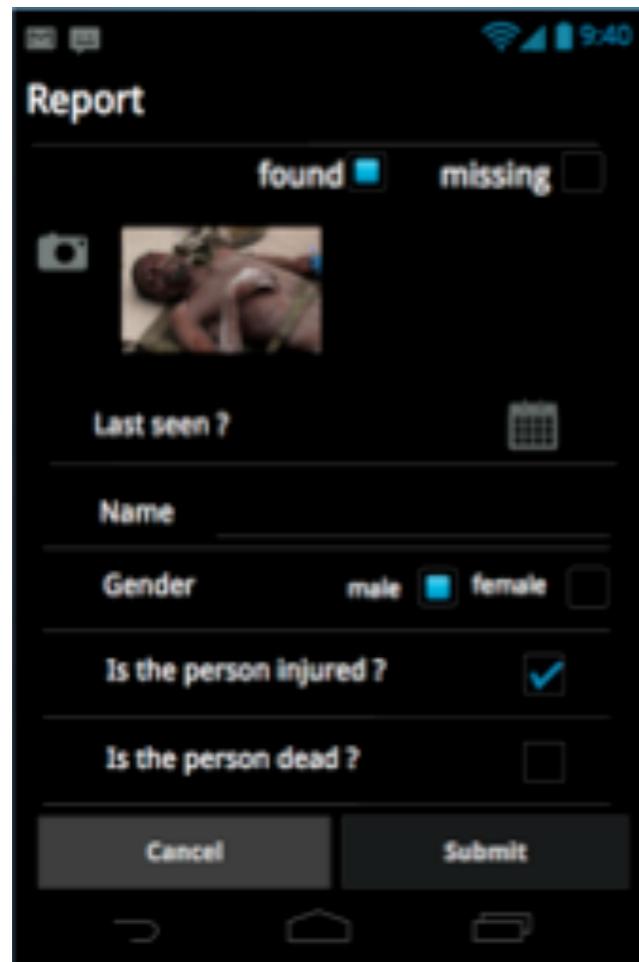
# Framework in Practice: Building Your App



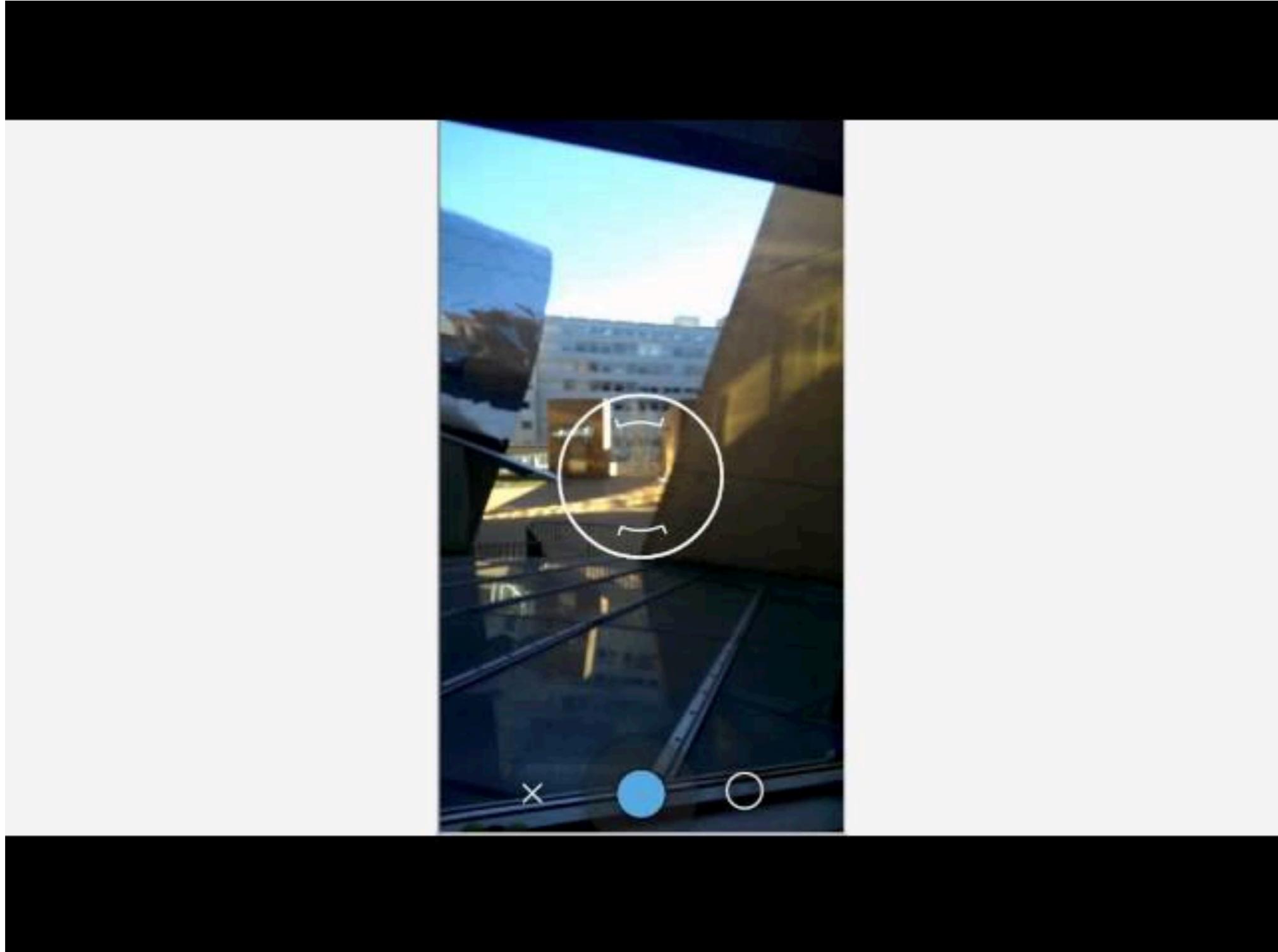
# Example App: Restoring Family Links



# Example App: Restoring Family Links



# Example App: WeReport



# Example App: WeReport

---

- **Allows users to submit reports about felled trees, down power lines, flooded roads, etc.**
  - Users are able to take pics, annotate them, and assign an appropriate category (such as flooded roads)
  - Users can also subscribe to receive real time updates of reports about different categories in their area
- WeReport: <https://www.youtube.com/watch?v=3gGJrMHasg>

# Current Development

---

- **Offline tolerance**
  - Web requests are cached and replayed
  - Wifi Direct as another channel
- **Usability of Linked Data**
  - Making Linked Data easier to use and deploy in mobile contexts
- **Template apps**
  - Reuse parts of apps easily
- **Privacy**
  - Allow data to be collected, stored, and used in a privacy aware manner

# Trying it out

---

**Try it:**  
[punya.mit.edu](http://punya.mit.edu)

**Contact us:**  
[punya-info@csail.mit.edu](mailto:punya-info@csail.mit.edu)

# Project Team



WeiHua Li  
MEng @ MIT



Julius Adebayo  
SM @ MIT



Eduardo Leon  
Undergrad @ MIT



Anubhav Jain  
Undergrad @ MIT



Andrew McKinney  
Tech Lead @ MIT App Inventor



Carlos Castillo  
QCRI PI



Patrick Meier  
QCRI PI



Lalana Kagal  
MIT PI