

# Master Thesis Presentation

## **iStuff Mobile: Rapidly Prototyping Novel Interactions for Mobile Phones**

### **Presenter:**

**Faraz Ahmed Memon**

**MSc. Software Systems Engineering**

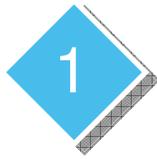
### **Advisors:**

**Prof. Dr. Jan Borchers**

**Prof. Dr.-Ing Stefan Kowalewski**

# Agenda

- 1 Introduction and Motivation
- 2 Background
- 3 iStuff Mobile
- 4 Prototyping using iStuff Mobile
- 5 Related Work
- 6 Conclusion



## Introduction and Motivation

- Importance of rapid prototyping of mobile phone interactions
- Introduction to iStuff Mobile

# Importance of prototyping of mobile phone interactions

- ❖ Mobile phones being used for more than just making/receiving calls
- ❖ Running prototypes needed before making crucial design decisions
- ❖ Introduction of user involvement
- ❖ Valuable user feedback
- ❖ Exposure of potential future system enhancements.



# Introduction to iStuff Mobile

- ❖ Software toolkit allowing rapid prototyping of mobile phone interactions
- ❖ Enables sensor enhanced interactions
  - Externally attached physical sensors
- ❖ Uses readily available mobile phones
- ❖ Allows prototyping of ubiquitous computing interaction
  - Interactive spaces
- ❖ Contribution of this master thesis into iStuff Mobile:
  - iStuff Mobile smartphone background application
  - Integration of Smart-Its sensor network platform into iStuff Mobile

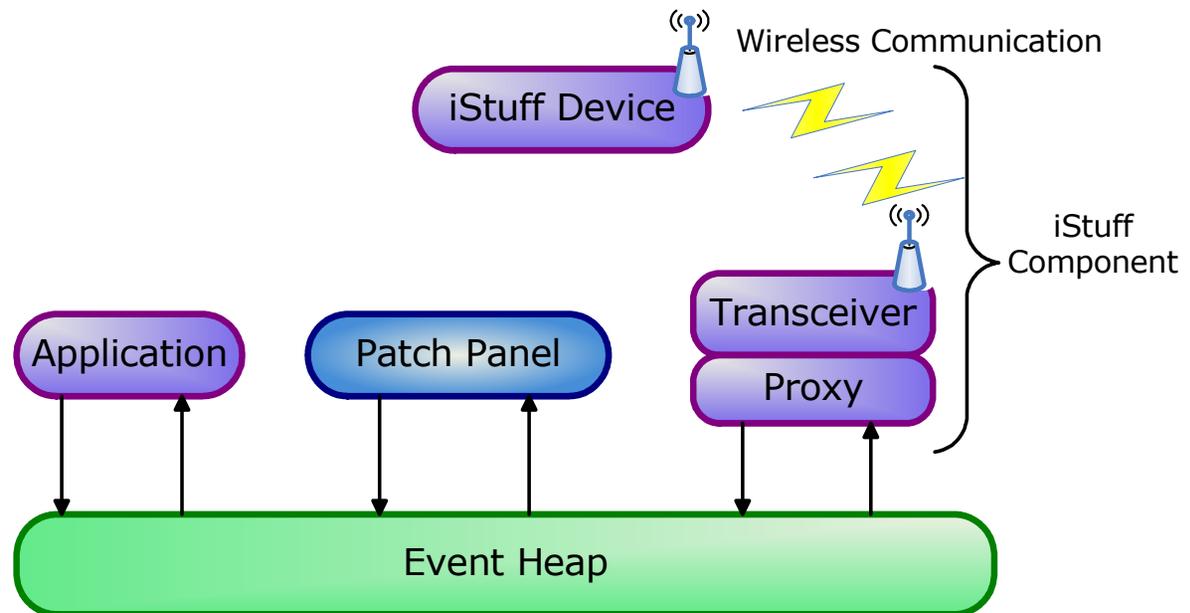


## Background

- iStuff Toolkit
  - iStuff Architecture
  - Event Heap
  - iStuff Components
  - Patch Panel
- Smart-Its sensor network platform
  - Smart-Its Architecture
  - Core and Sensor Boards
  - Smart-Its Communication

# iStuff Toolkit Architecture

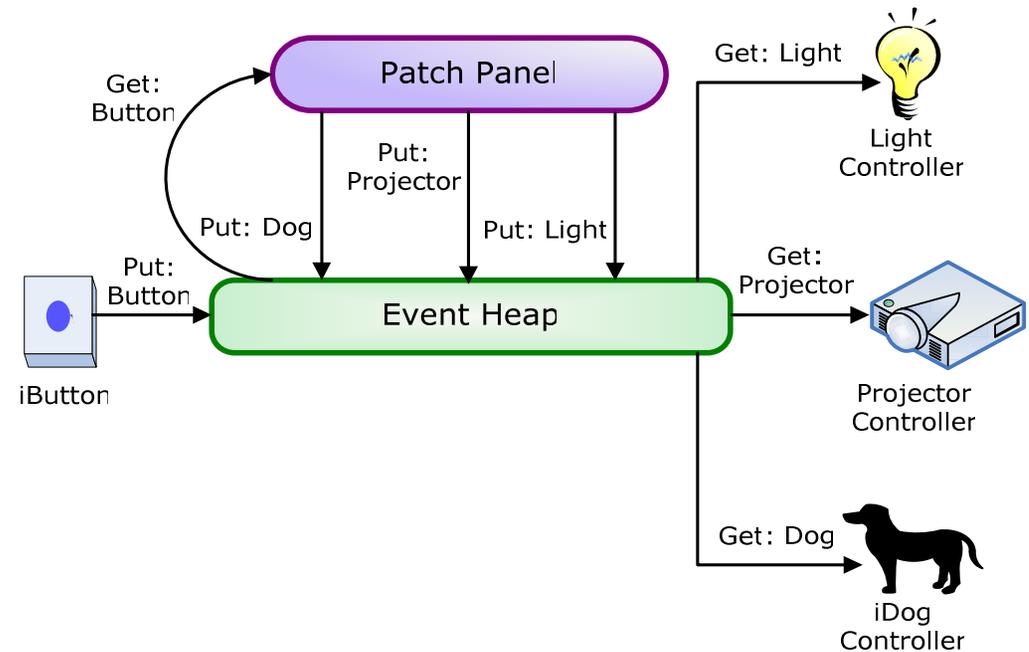
- ❖ iStuff Toolkit
  - Toolkit for design and exploration of novel interactions
- ❖ iStuff Architecture
  - Event Heap
  - iSuff Component
  - Patch Panel



# Patch Panel

- ❖ Ubicomp environment
  - Devices/services added frequently
  - Communication cannot be anticipated

- ❖ Patch Panel
  - Retargeting event flow
  - Event transition or finite state machine



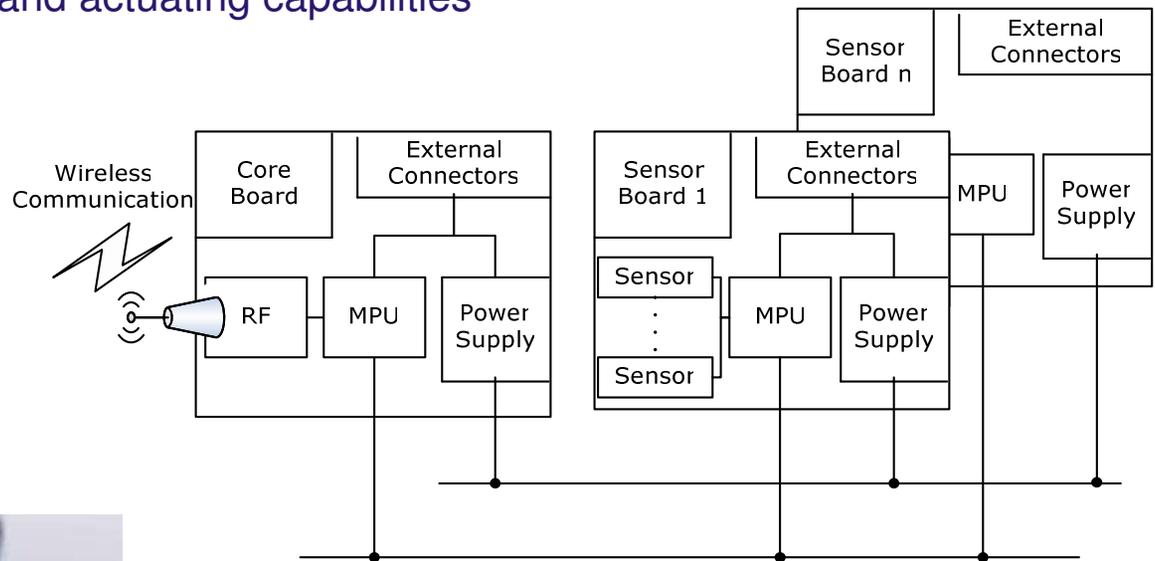
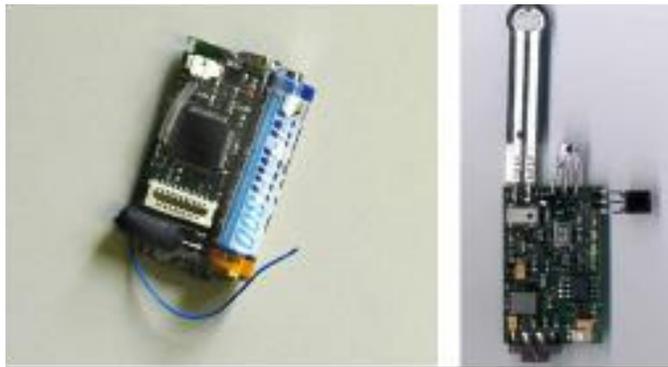
# Smart-Its Architecture

## ❖ Smart-Its (Particle Computer)

- Small scale low powered embedded systems
- Add sensing, processing and actuating capabilities
- Two independent boards
- Wireless communication

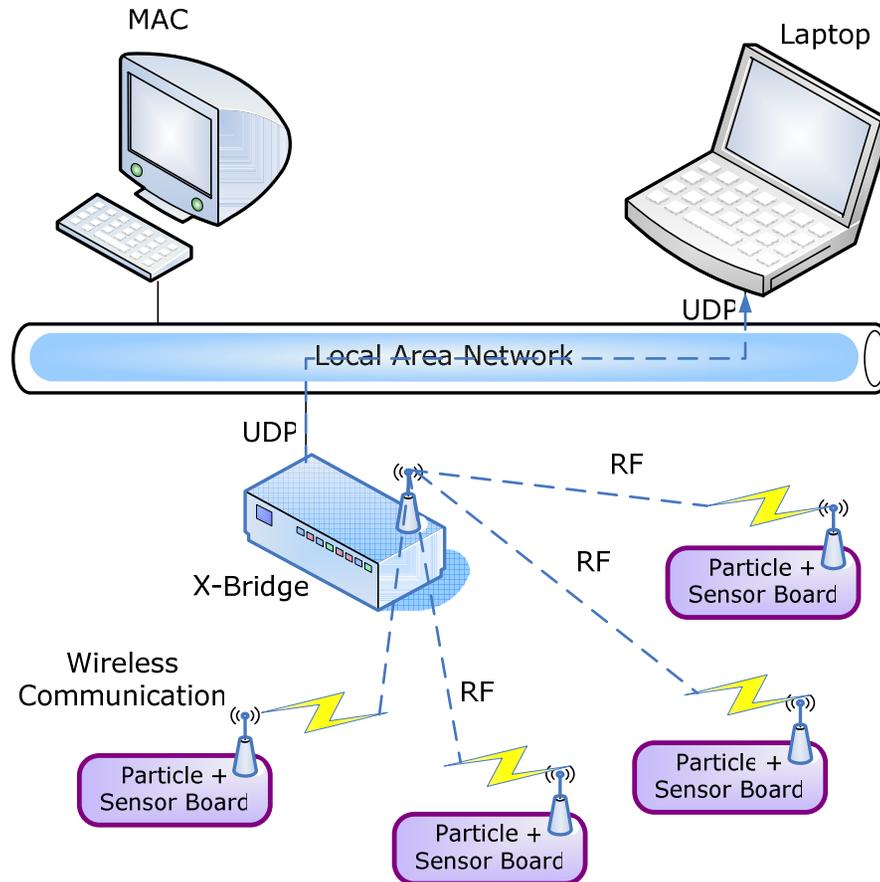
## ❖ Smart-Its Architecture

- Core board
- Sensor board
- Uptil 16 sensor boards



# Smart-Its Communication

## ❖ Smart-Its Communication





## iStuff Mobile

- iStuff Mobile Architecture
- Particle Framework
- iStuff Mobile Proxy
- iStuff Mobile smartphone background application
- iStuff Mobile smartphone foreground application
- Visual programming support

# iStuff Mobile

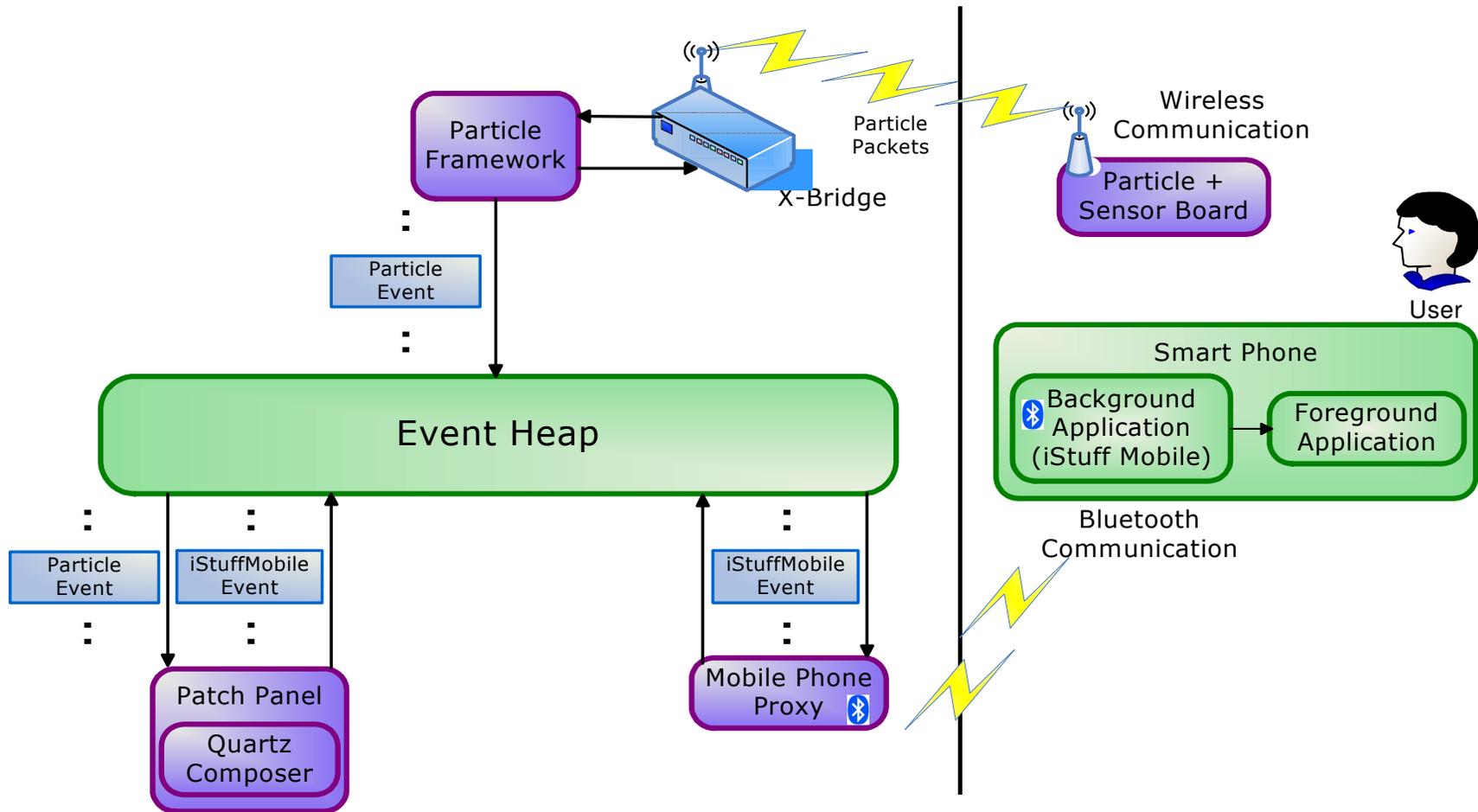
## ❖ iStuff Mobile

- Extends the iStuff Toolkit
  - Incorporates Smart-Its platform through proxy strategy
  - Proxy for mobile phone
  - Provides Symbian series 60 mobile phone application

## ❖ iStuff Mobile Architecture

- Compound prototype architecture
- Components spread accross the room
- Communication through Event Heap

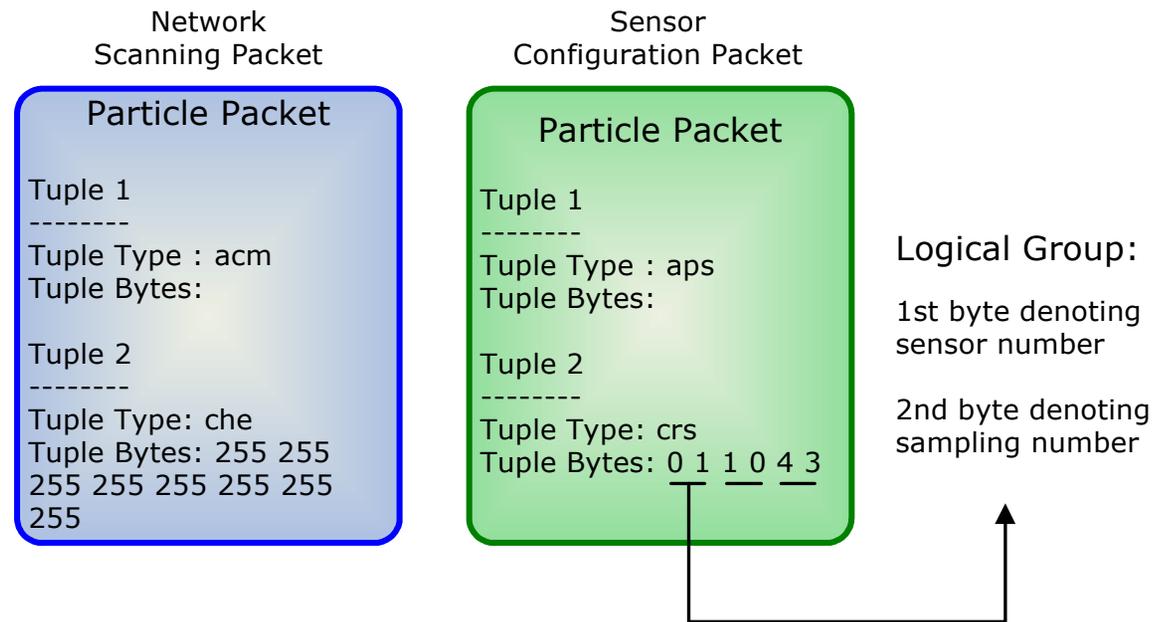
# iStuff Mobile Architecture



# Particle Framework

- ❖ Particle Framework ([Screenshot](#))
  - Cross-platform GUI
  - Scanning for Particles in network
  - Configuration of sensors
  - Reception of sensor data
  - Posting sensor data to Event Heap
- ❖ Scanning network for Particles
  - Transmission of HELLO packet
  - Reply of HELLO packet
  - Addition of Particle Id to the list on GUI

# Scanning and Configuring Particles



## ❖ Configuring Particle sensor board

- Transmission of configuration packet
- Acknowledgment if configuration successful
- Sensor numbers
- Sampling numbers

# Sensor Data Transmission

## ❖ Sensor Data Transmission

- Particle packets
- Particle events
- Filtering process
- Conversion from particle packet to particle event

# iStuff Mobile Proxy

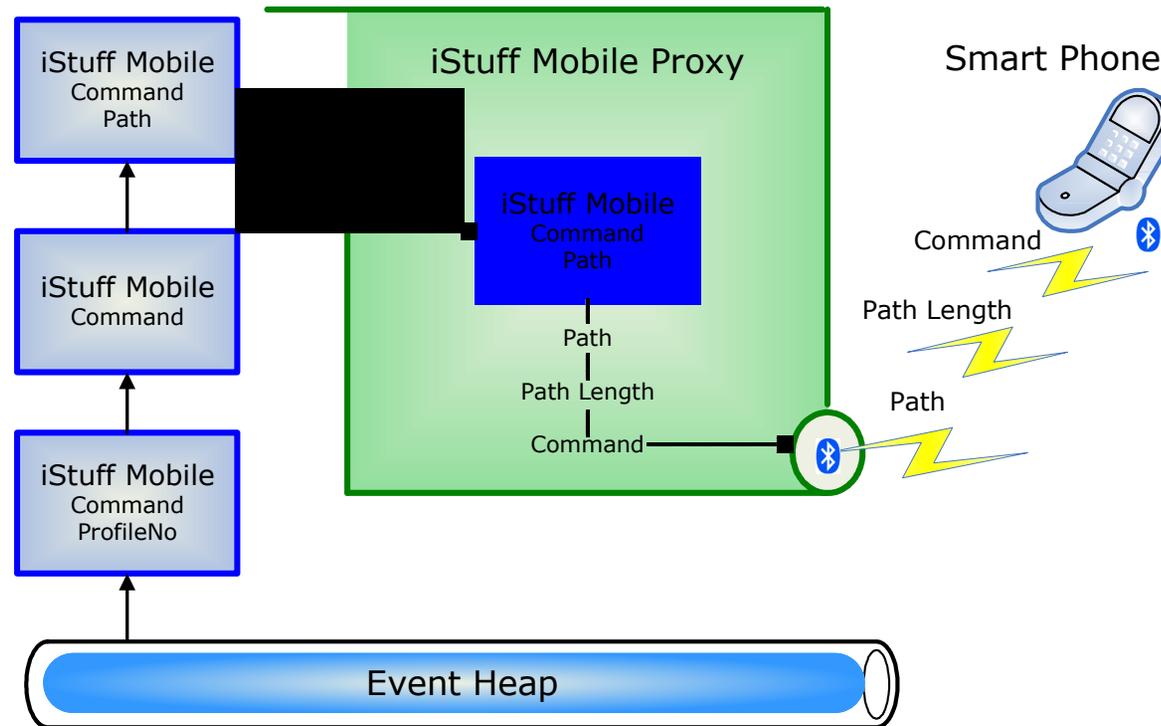
## ❖ iStuff Mobile Proxy

- Command-line program
- Communication with mobile phone
- Relaying of events to the mobile phone
- Relaying of user activity to the Event Heap

## ❖ Event relaying process

- Register for “iStuff Mobile” events with “Command” field
- Events with “Command”
- Events with “Command” and “Path”
- Events with “Command”, “Repeat”, “Code” and “ScanCode”
- Events with “Command” and “ProfileNo”

# Event Relaying Process



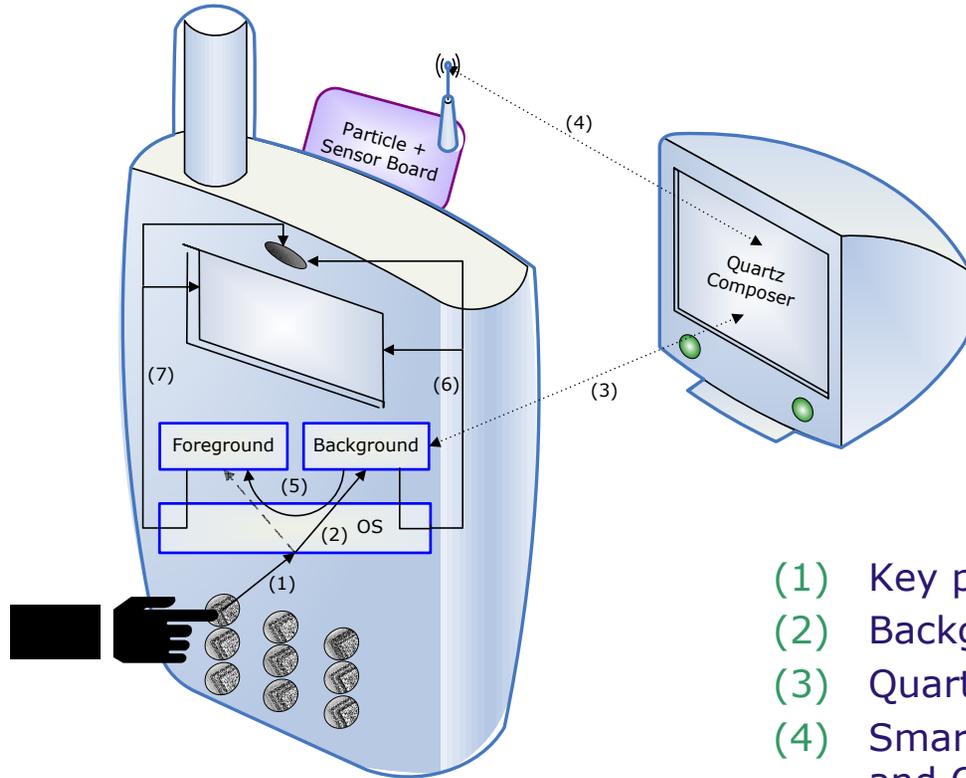
## ❖ User activity relaying process

- User activity in form of key presses
- Listen for incoming commands
- Command followed by ascii value of key and type value
- Post "iStuff Mobile" event with "KeyCode" and "Activity" fields

# iStuff Mobile Smartphone Application

- ❖ iStuff Mobile smartphone application
  - Background application
  - Foreground application
- ❖ Smartphone background application
  - Provided by the framework
  - Allows remote execution of commands
  - Interaction the foreground application
  - Interaction with the mobile phone OS
  - Interaction with iStuff Mobile Proxy

# Smartphone Background Application



- (1) Key press sent to OS
- (2) Background application intercepts key press
- (3) Quartz composer gets notified of key press
- (4) Smart-Its reporting sensor information to QC and QC issues high-level command to the background application
- (5) Key press sent to foreground application
- (6) System output
- (7) Foreground application response

# Smartphone Background Application

- ❖ Feature of smartphone background application
  - Bluetooth communication
  - Sound playback
  - Vibrator control
  - Key capture capability
  - Foreground application key simulation
  - Launch external application
  - Close external application
  - Profile control
  - Backlight control
  - Run application in background
  - Camera control
- ❖ Current version supports Symbian Series 60 smartphone
- ❖ Windows Mobile 5.0 is a good candidate ([Survey](#))

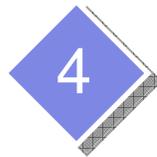
# Smartphone Foreground Application

## ❖ Foreground application

- Visible to user
- Responds to smartphone background application
- Default series 60 application
- Custom designed applicaion
  - Macromedia Flash Lite
  - Python for Series 60
  - J2ME
  - Symbian C++

## ❖ Visual Programming Support

- Apple's Quartz Composer
- Patch Panel Mappings



## Prototyping with iStuff Mobile

- Recreating inspiring mobile phone interaction
- Ubiquitous computing prototyping scenarios

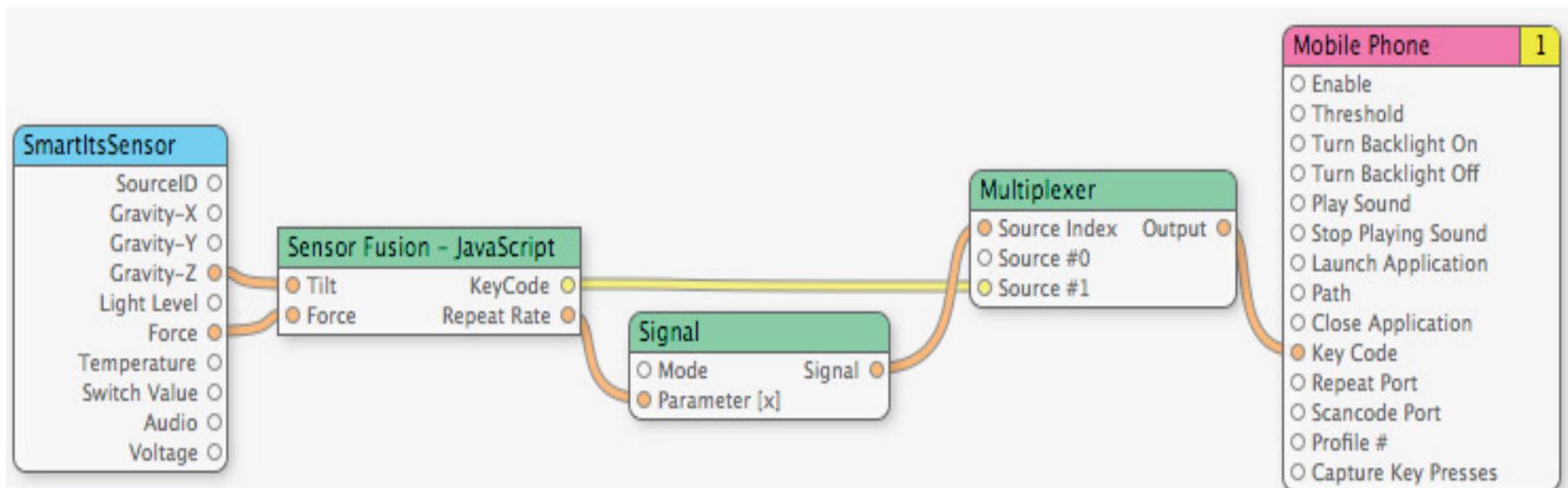
# Prototyping with iStuff Mobile

- ❖ Recreating inspiring mobile phone interactions
  - Tilt scrolling
  - Changing ringing profile
  - Tilt typing
- ❖ Ubicomp prototyping scenarios
  - Multi-screen presentation control
  - Keyboard input redirection
  - Speech text

# Tilt Scrolling

- ❖ Tilt scrolling [Harrison et al., 1998]
  - Using tilt to scroll through list
  - Speed of Scroll
  - Originally created using
    - Palm Pilot
    - Pressure sensor
    - Tilt sensor

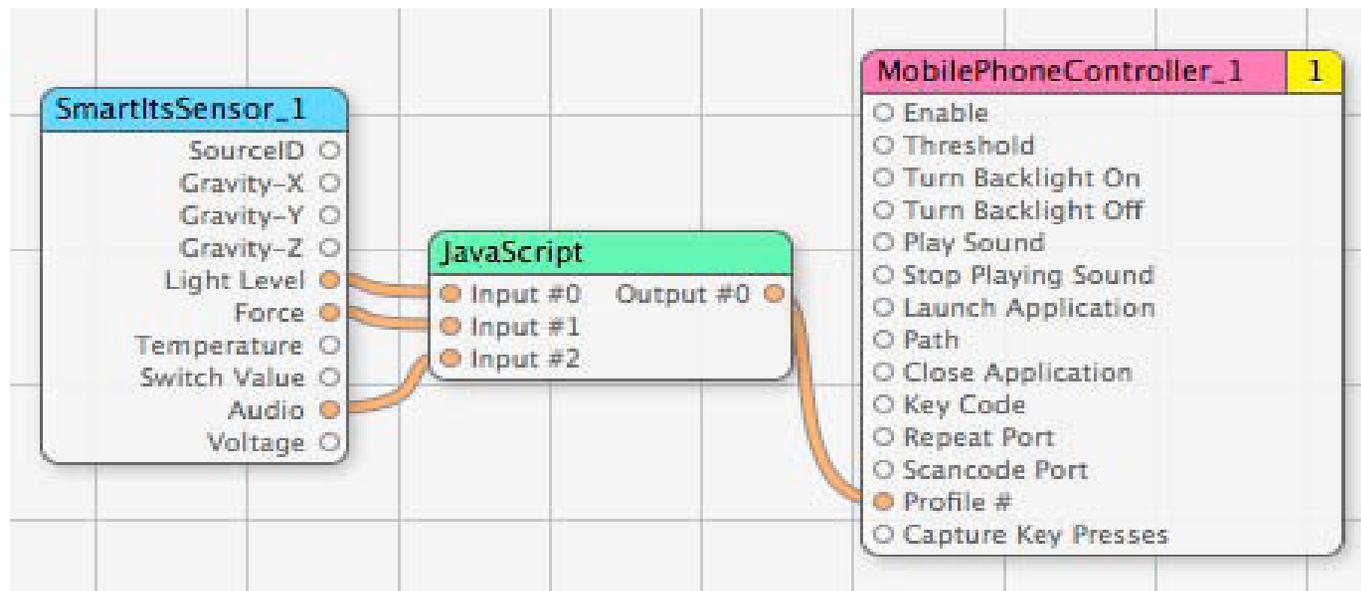
- ❖ Recreated using
  - Nokia Series 60 mobile phone
  - Smart-Its pressure sensor
  - Smart-Its acceleration sensor
  - Foreground application key simulation



# Changing Ringing Profile

- ❖ Changing ringing profile [Schmidt et al., 1999]
  - Use context to change user profile
  - Definition of profiles
  - Originally created using
    - Nokia phone
    - TEA board

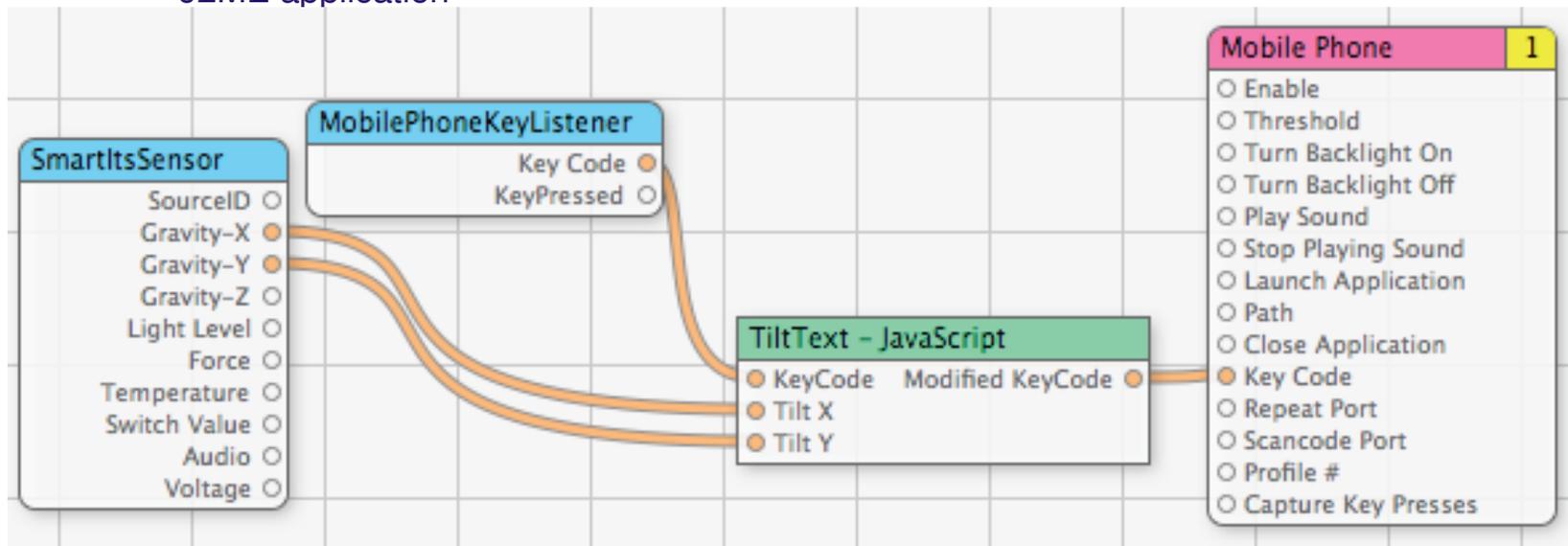
- ❖ Recreated using
  - Nokia Series 60 mobile phone
  - Smart-Its pressure sensor
  - Smart-Its light sensor
  - Smart-Its Audio sensor



# Tilt Typing

- ❖ Tilt typing [Wigdor and Balakrishnan, 2003]
  - Use tilt + key press to type a character
  - Faster than multi-tap
  - Comparable to T9 dictionary
  - Originally created using
    - Motorola i95cl phone with external sensors
    - J2ME application

- ❖ Recreated using
  - Nokia Series 60 mobile phone
  - Smart-Its acceleration sensor
  - Mobile phone key capture
  - Foreground application key simulation



# Ubicomp Prototyping Scenarios

## ❖ Multi-screen presentation

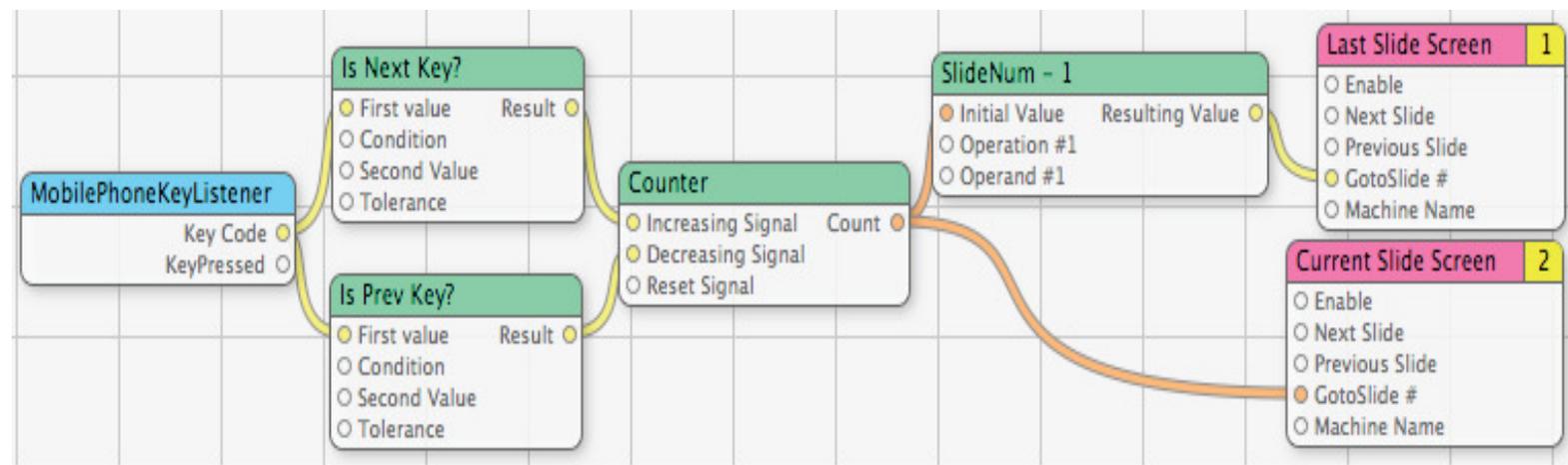
- Two displays showing current and previous slides
- Key press on mobile phone to navigate
- Proxy running on two separate machines
- Patch panel mapping

## ❖ Keyboard input redirection

- Java “TextEventEngine” software proxy
- Foreground application key simulation

## ❖ Speech text

- Keyboard input redirection
- iListen





## Related Work

- Phidgets
- Calder toolkit
- The TEA project
- D.Tools
- ContextPhone



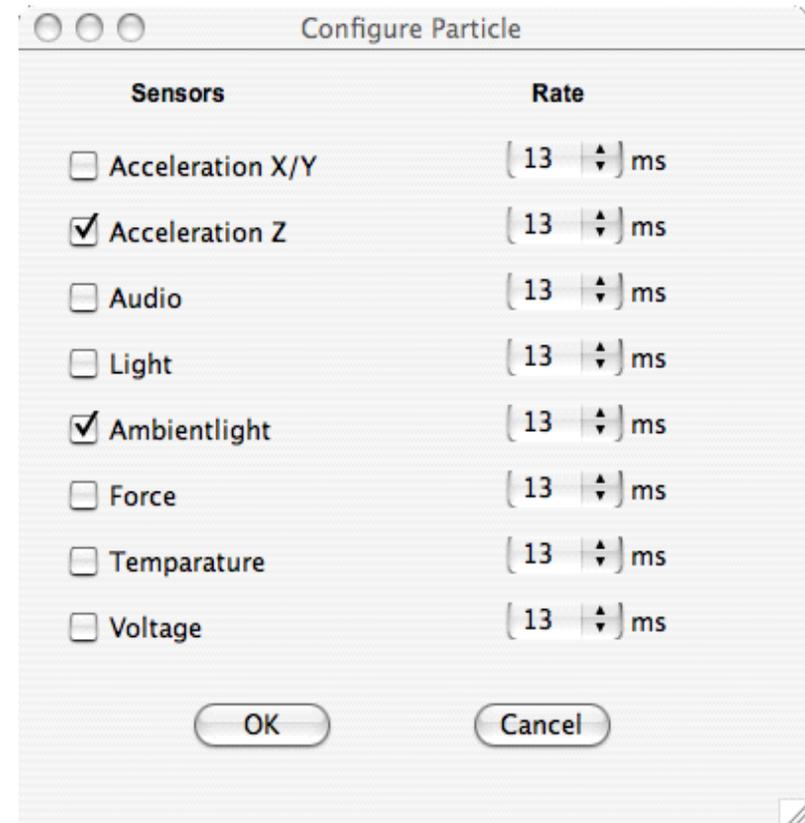
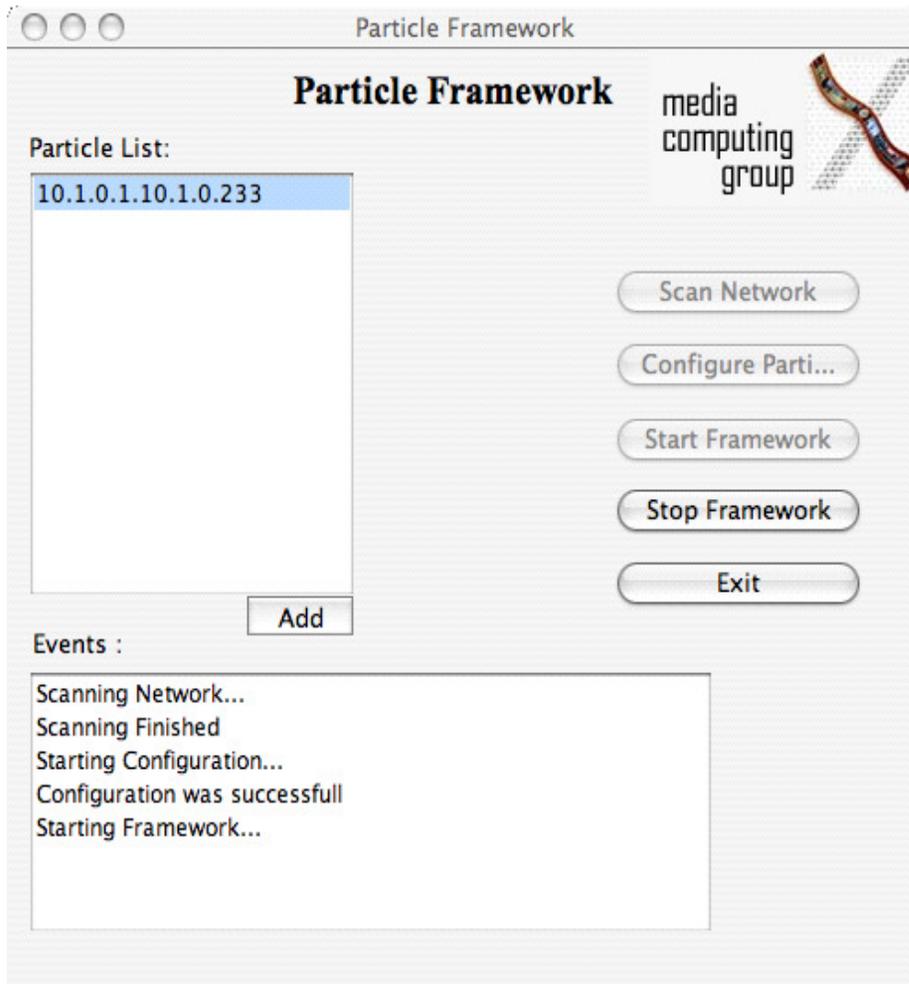
## Conclusion

- Future work
  - New background application features
  - Integrate Sweep and Point & Shoot application
  - Port to Window Mobile 5.0 and Linux phones
  - Provide Inter-Process communication between foreground and background application
- Demo
- Questions?



# Thank You !

# Particle Framework



# Particle Framework

Sensor Number	Sensor Name
0	Acceleration X/Y axis
1	Acceleration Z axis
2	Audio
3	Light
4	Ambient Light
5	Force
6	Temperature
7	Voltage

Sampling Number	Sampling Rate
0	13 ms
1	26 ms
2	52 ms
3	104 ms
4	208 ms
5	416 ms
6	832 ms
7	1664 ms
8	3328 ms
9	6656 ms
10	13312 ms
11	26624 ms
12	53248ms
13	106496 ms
14	212992 ms
15	425984 ms



# Smartphone Background Application Survey

	Bluetooth Communication	Sound Playback	Vibrator Control	Key Capture Capability	Foreground Application Key Simulation	Launch External Application	Close External Application	Profile Control	Backlight Control	Run Application in Background	Camera Control
<b>Symbian Series 60</b>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <sup>a</sup>	<input checked="" type="checkbox"/> <sup>b</sup>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <sup>c</sup>
<b>Java (J2ME)</b>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/> <sup>d</sup>		<input type="checkbox"/> <sup>e</sup>					<input type="checkbox"/>
<b>Windows Mobile 5.0 SmartPhone</b>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- a. Profile change implemented using "External Application Launch", "Key Press Simulation" and "External Application Termination"
- b. Only turn ON functionality
- c. Sweep and Point & Shoot implemented as a separate application
- d. Available only when Java application in foreground
- e. Available only when Java application in same package

