3dRAWE

Ryan Cahoon, Mike Ralph, Yiling Tay, Yumin Wong
### Priya Narasimhan, Rajeev Gandhi

Motivation

People today have the chance to experience augmented reality through current smart phone apps but they are usually fixed in content and non-interactive.

Our prototype will provide users with a rich multimedia environment:
• Draw 3D objects in space, in front of the Kinect
• View them on different surfaces conveniently on their phone
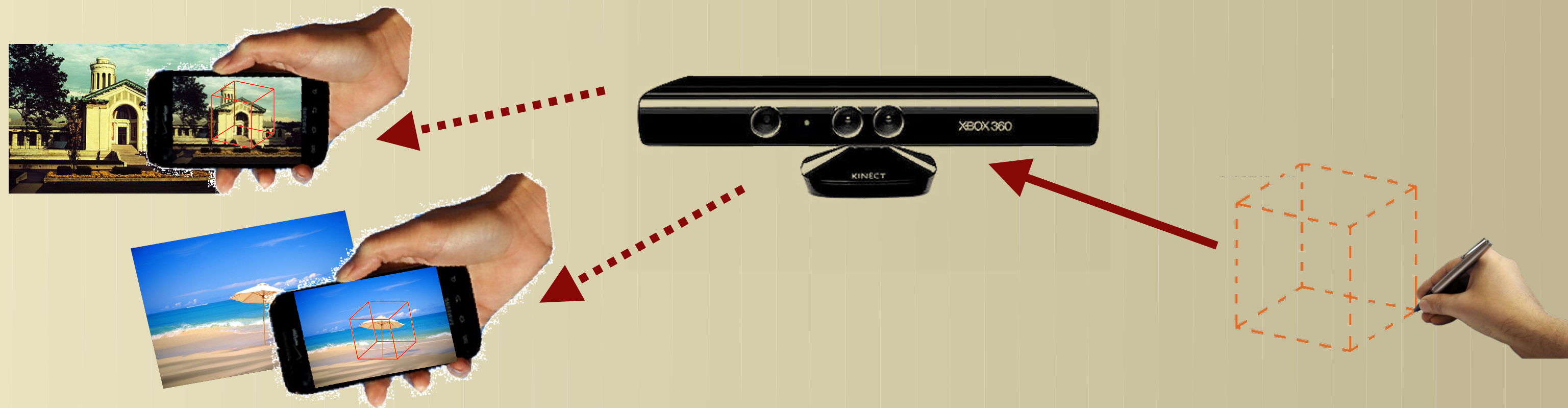
Architecture
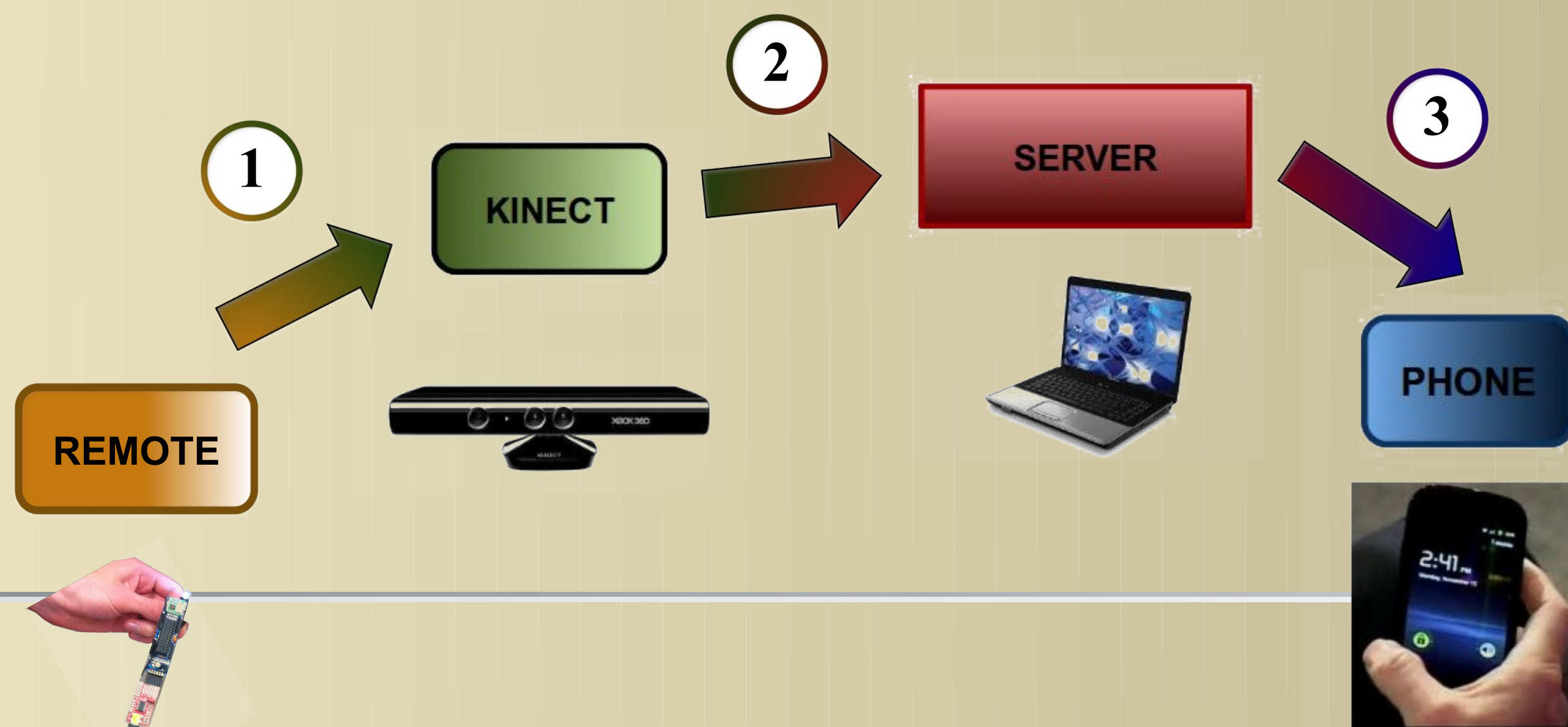
1. The Kinect detects the bright LED light of our remote/drawing tool, reads the points and packs them into data packets.

2. Data packets are sent from the Kinect and read by the Server via USB Connection.

3. The server forwards the packets to the Android phone via Wi-fi connection. The phone reads and unpacks the data packets into points for rendering on its screen.

System Components

Hardware
• Android: T-Mobile G1, Firmware version 1.6, updated to 2.1
• Drawing Tool/Remote Control: JeeNode Main MCU, JeeLink Server Interface, JeeLabs Power Supply Board, USB BUB Board, LEDs
• Kinect

Server Interface

Power Supply Board

Main MCU

USB BUB Board
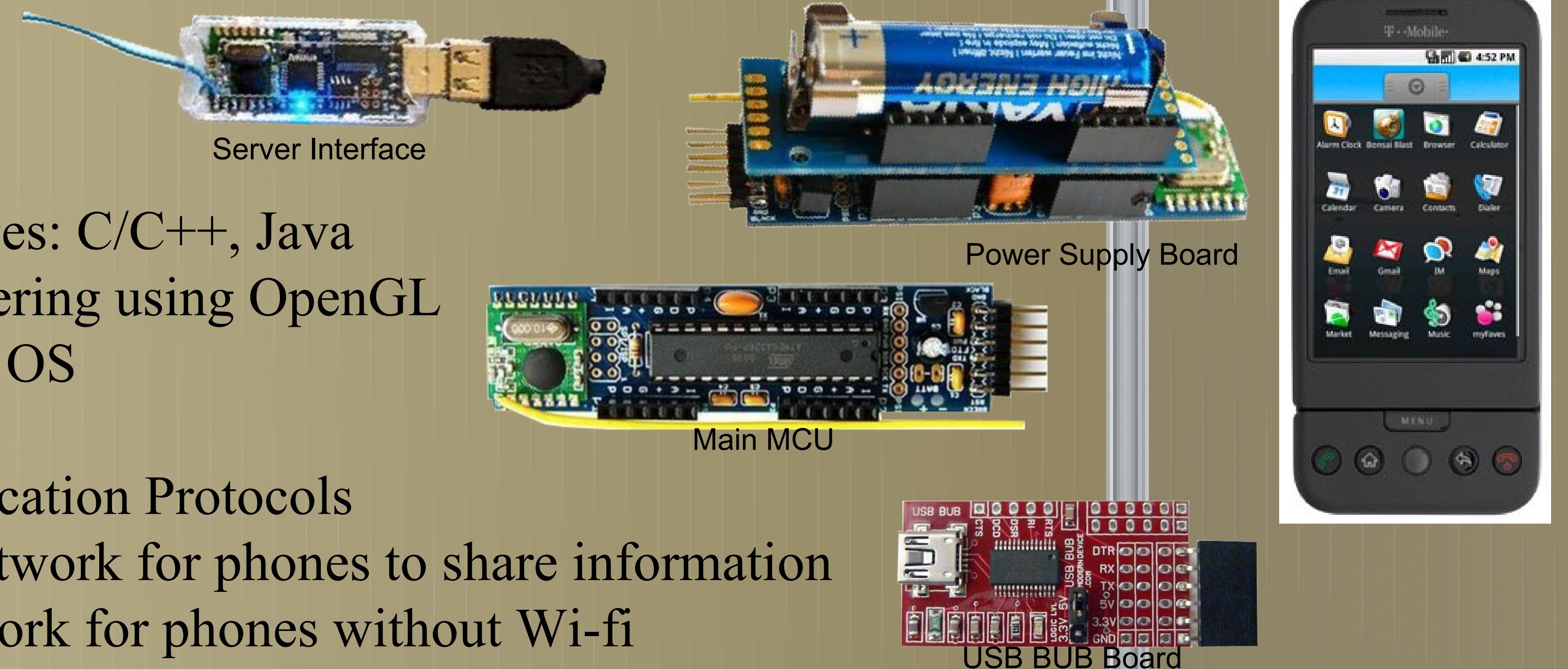
Software
• Languages: C/C++, Java
• 3D rendering using OpenGL
• Android OS

Communication Protocols
• Wi-fi network for phones to share information
• 3G network for phones without Wi-fi
• USB communication between Kinect and server

Results

General Latency
• Tracking phone location: ~ 2.3 fps
• Graphics rendering: ~ 18.4 fps
• Kinect: not a limiting factor since it is connected via USB

Latency in forwarding of points from the Kinect to the Android phone
• Initial spike: ~ 4.8s lag due to the initial set-up of the Android phone and commencement of the rendering of points
• Constant lag: plateaus at ~1.9s due to sending of data packets over the network

Latency in Forwarding Data Packets

Time Lag (s)

Data Packet Seq Id