

Cube SDK

Building an Application for the MS Cube

Stimulant (why are we here?)

- We specialize in building interactive installations
- Worked with MS on the Surface
- Have developed many Kinect 1.0 applications
- Created the first app for the Cube (dBcube)

A little about the Cube



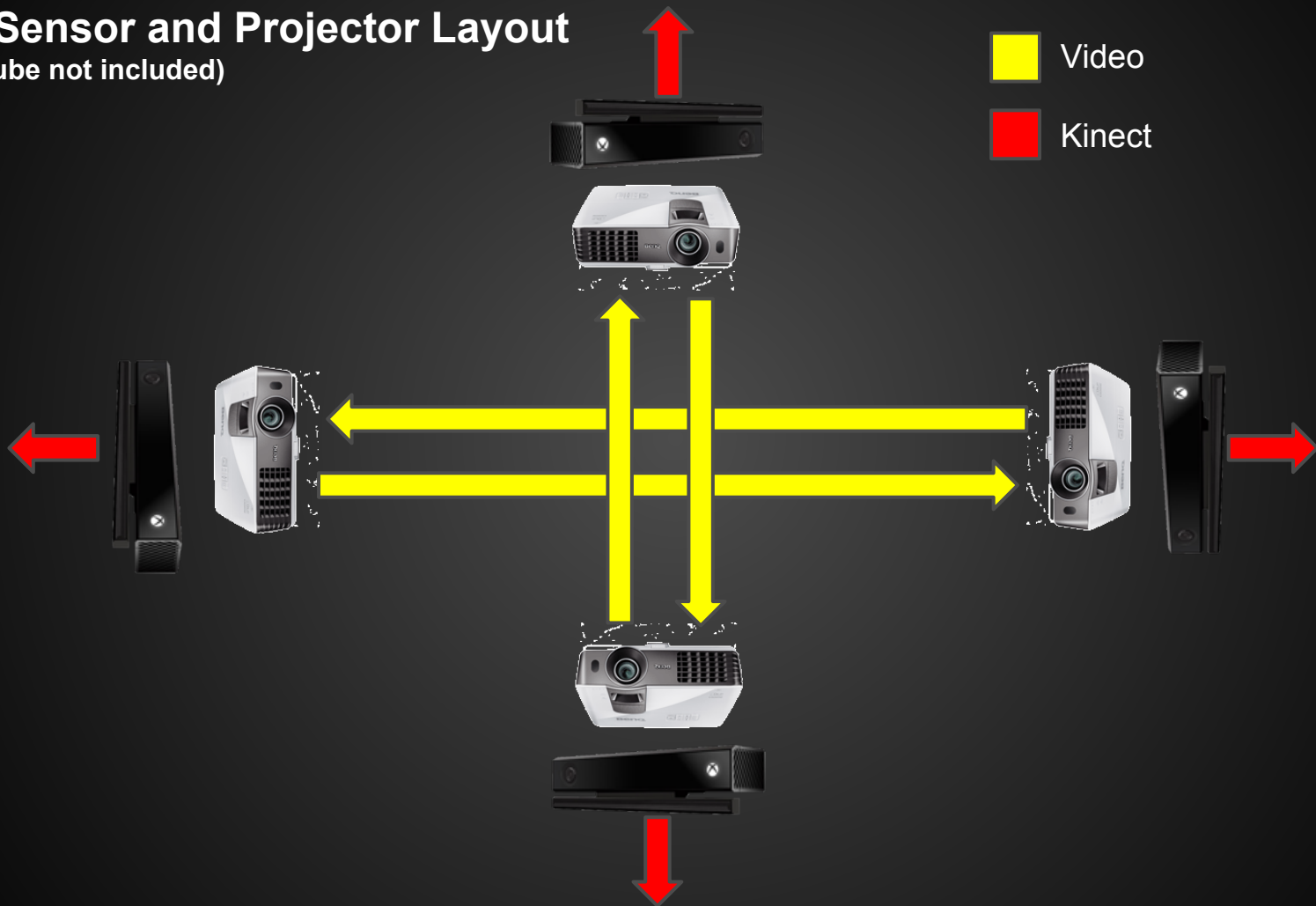
The CUBE

- 5 Intel NUC PCs
- 4 Kinect V2.0
- 5 Benq Projectors (1024x768 clipped to 768x768)
- 4 Acrylic rear projection screens (each 4'x4')


Cube Sensor and Projector Layout

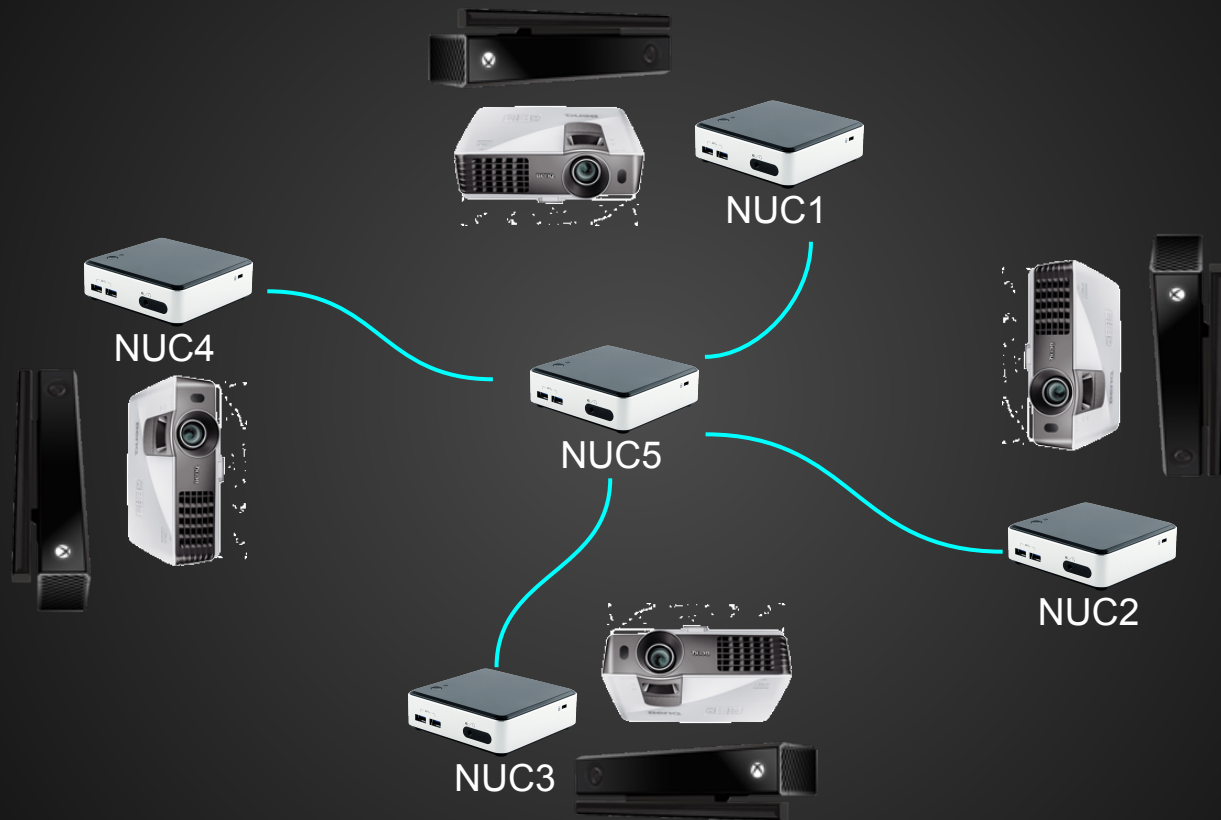
(top of Cube not included)

Video
Kinect



Cube PC Layout

 Network Connection



Building an application for the Cube

- Challenge is sending data and creating a shared experience
- Many different topologies for Client/Server configuration depending on needs of application
- Need to decide early on what shared experience you are trying to create and what data you will need in order to create it
- Network bandwidth has it's limits

SDK Philosophy

- Allow you to send data anywhere you want with Kinect Transport application
- Provide samples and source code for consuming that data and visualizing it
- Some data can exist on one client and some data will probably need to be sent to a server to create a shared experience

Possible Cube Application Topologies

- No server, each side is a distinct experience (Cinder, Unity)
- Dumb client, server provides all state and visuals (NodeJS)
- Distributed system, server sends shared experience to each client to generate visuals (Cinder, Unity)

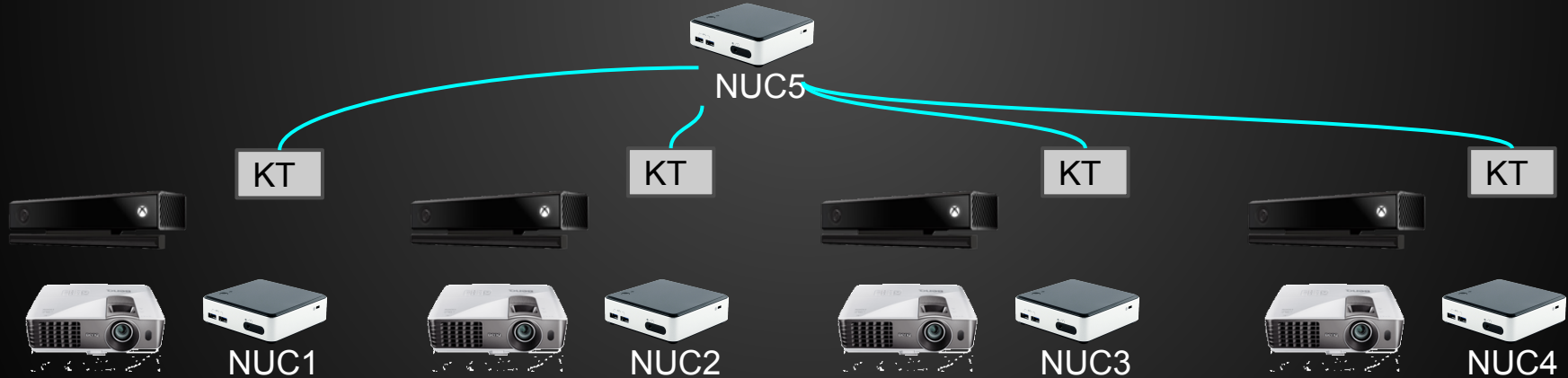
No Server

- Kinect Transport sends data to localhost (127.0.0.1)
- Data is visualized locally by Cinder or Unity app
- No shared state between sides



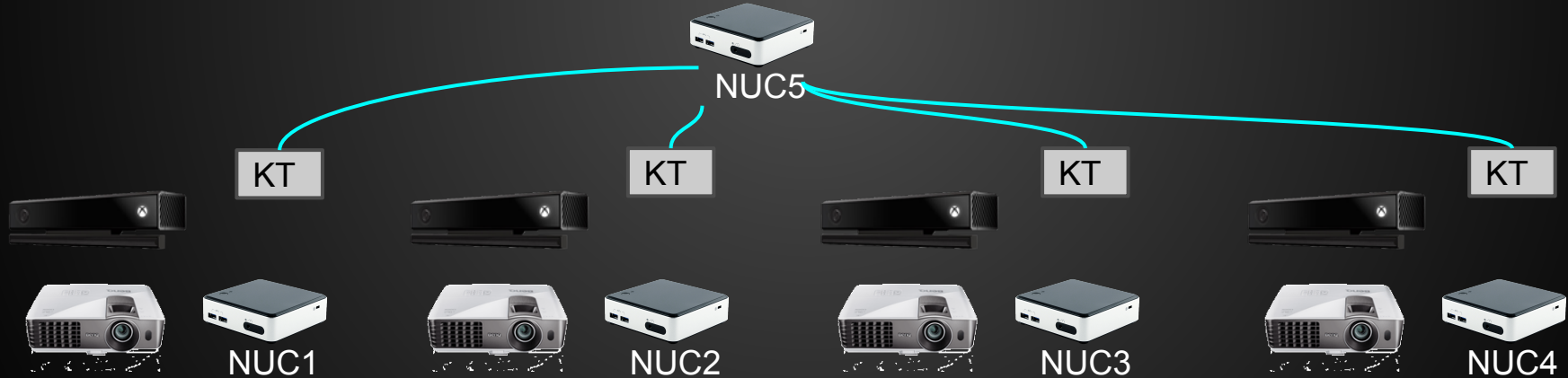
Dumb Client

- Kinect Transport sends data server (NUC5)
- Data is visualized on server as web page
- Browser on clients (NUC1-4) display web page



Distributed System

- Kinect Transport sends data server (NUC5)
- Shared state created on server
- State is sent back to clients to visualize



Kinect Transport

- Sends multiple streams of data (Bodies, Depth) to up to 4 destination hosts (can be localhost)
- Uses TCP/IP for networking and sends data as raw binary for speed and reliability
- Source code is provided and is easy to use/refactor
- C++ code for parsing, sending and receiving binary streams of data is re-used for Cinder (and eventually oF) sample apps

API Specification

- Description of binary data
- Helper source code for sending/receiving it
- General documentation

SDK Samples

- Samples in NodeJS, Cinder and Unity that demonstrate receiving and visualizing binary streams of data
- NodeJS sample could be used as either client or server (or both)
- Samples can be in any framework that supports receiving TCP/IP data (does not need to support Kinect 2.0 SDK)

Conclusion

- Github repo lives here: <https://github.com/stimulant/MS-Cube-SDK>
- Please try out the samples and give suggestions!
- SDK is a work in progress

Thanks!

Joel Pryde

@joelpryde joel@stimulant.com

Stimulant:

@stimulant <http://stimulant.com>