

*ECC Conference 2019*

# **Universal IoT Communications using Semantic I/O**

Brian Gormanly  
Brendan Kelly  
**Marist College**

<https://www.linkedin.com/in/gormanly>

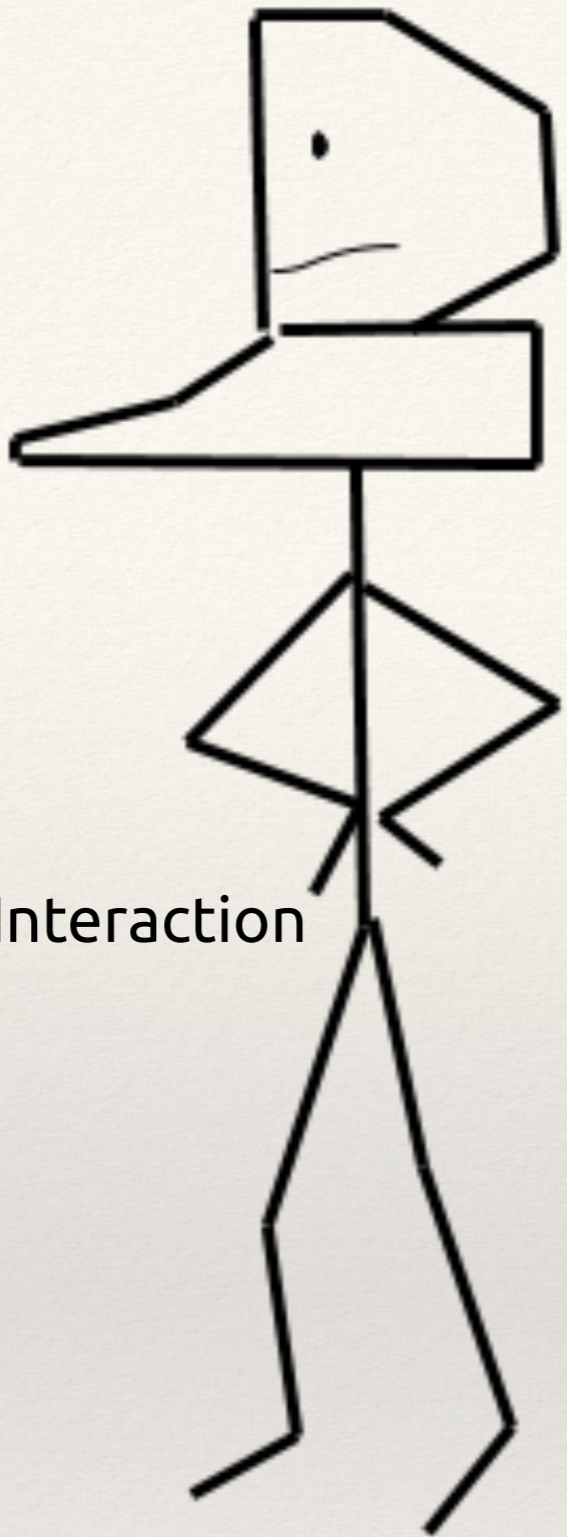






HCI

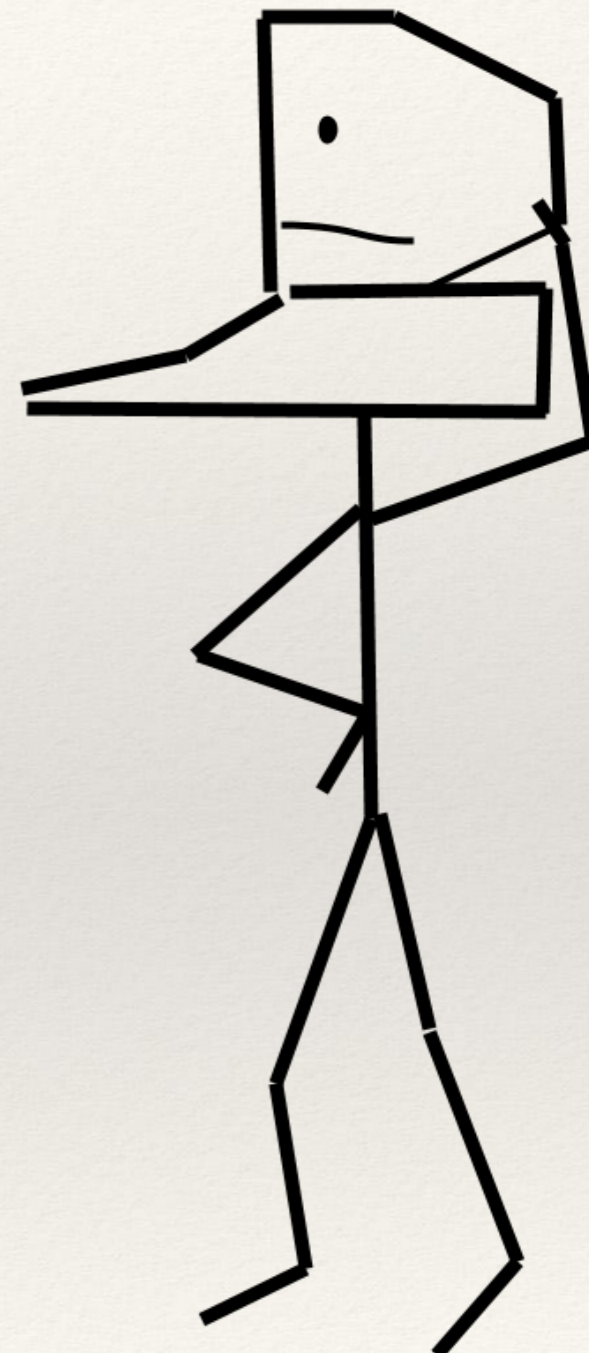
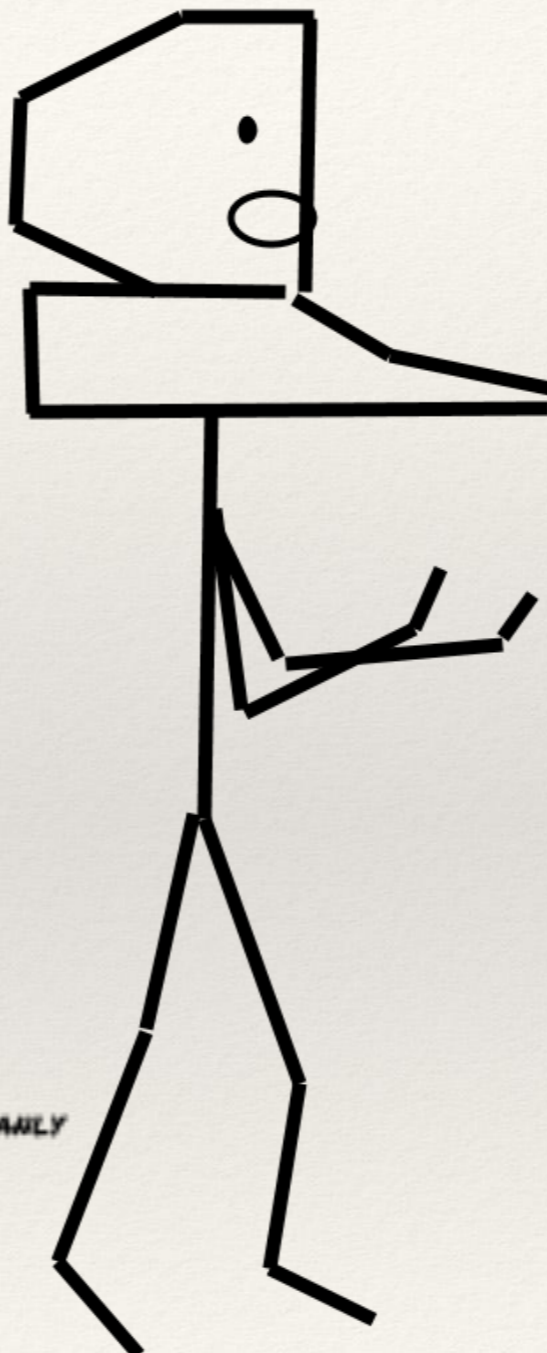
Human Computer Interaction



BRIAN GORMANLY

CCI ???

Computer - Computer Interaction

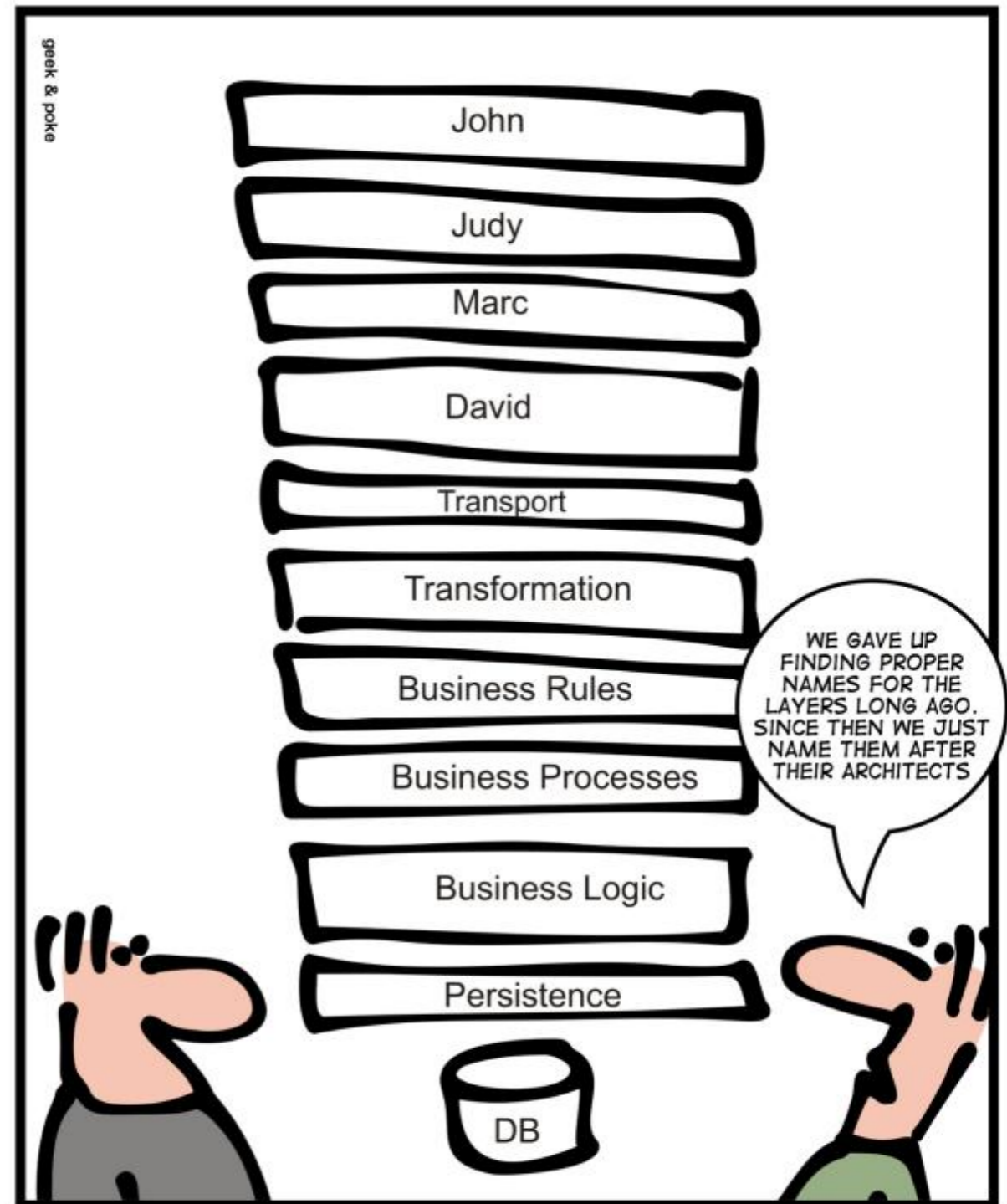


BRIAN GORMANLY





Application		
File system	GUI	Task management
OS		
Bootloader	Board support packages	Device drivers
Hardware		



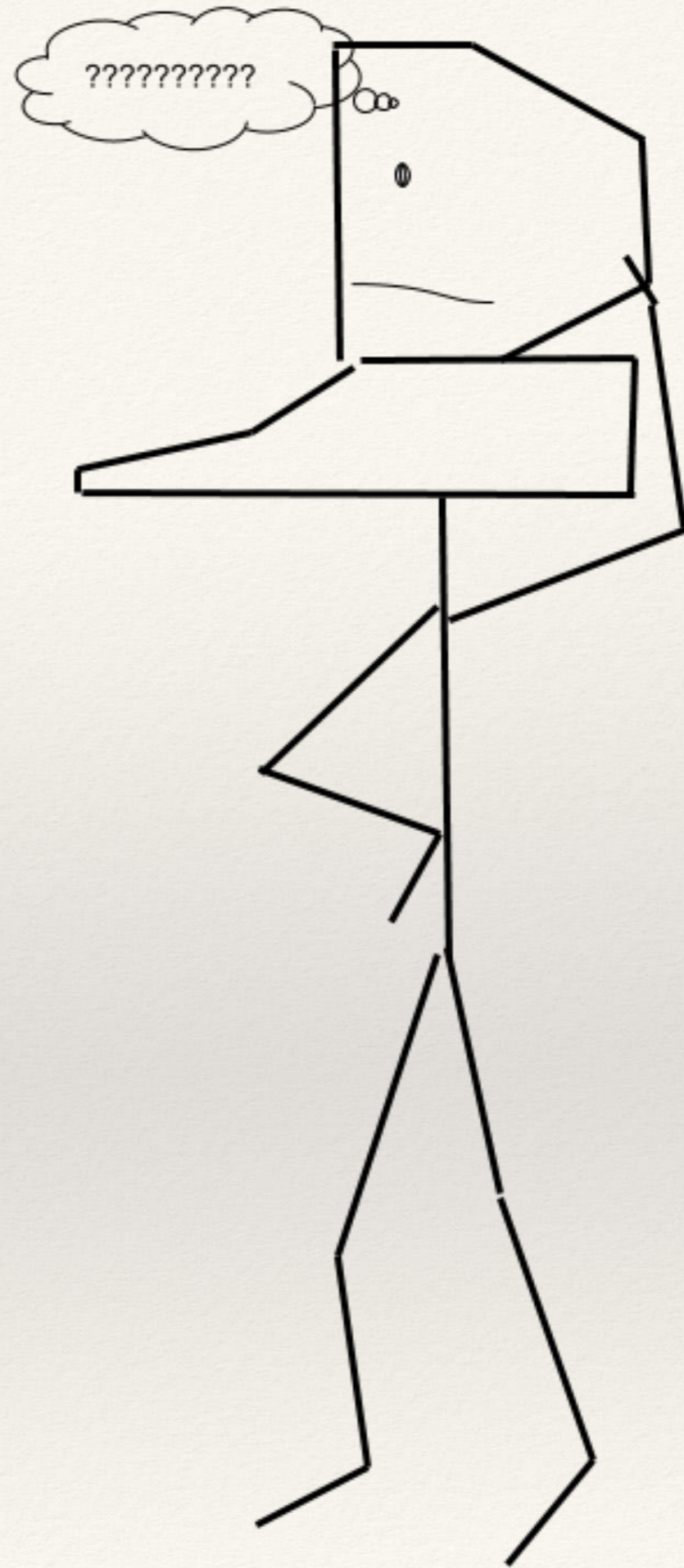
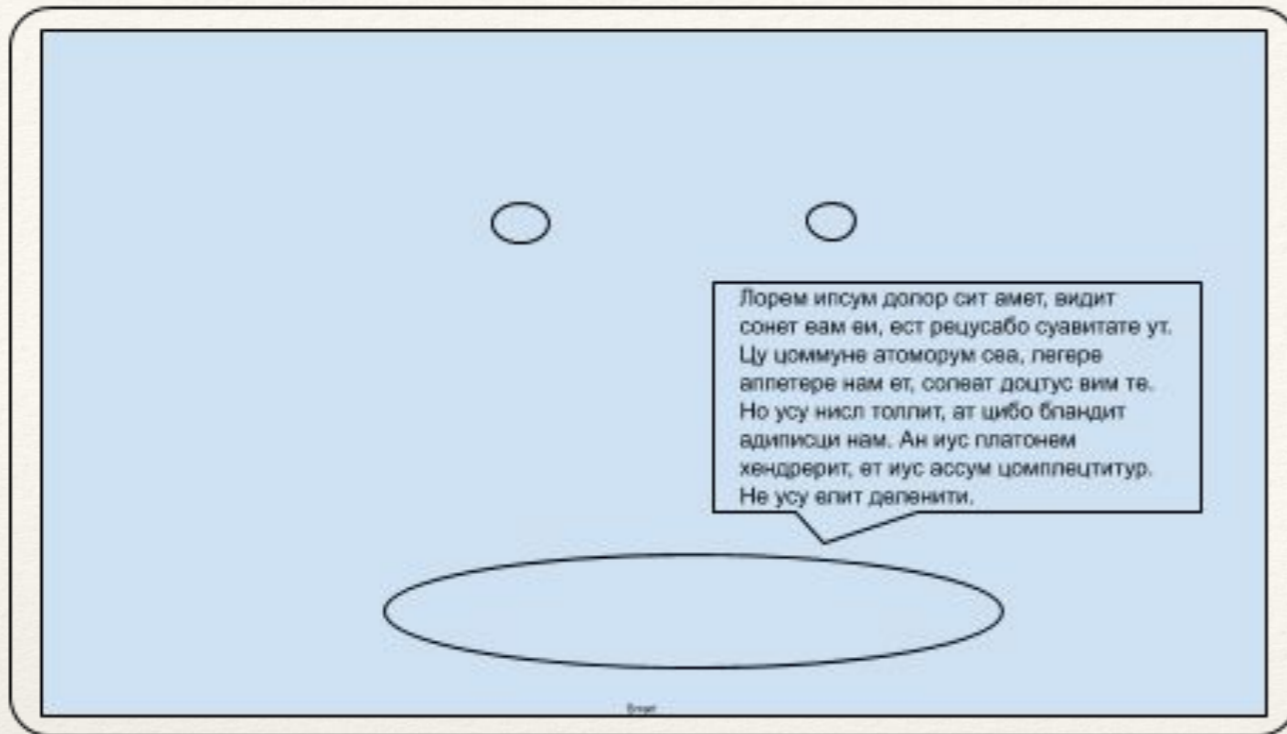
A GOOD ARCHITECT LEAVES A FOOTPRINT

\* Image Credit: <https://blog.jooq.org/2014/09/12/why-you-should-not-implement-layered-architecture/>









BRIAN GORMANLY



---

# What Can be Done?

---

If making devices smaller, less expensive and less power hungry means we have to compromise on complexity and complexity enables abstraction, is all lost?



---

# Interactions and Intent

---

An **Interaction** is some input or output expressed between people and systems or systems and systems.

**Intent** is a description of the goal of interaction.

What if we standardized ways of expressing interactions by describing intent **semantically**?

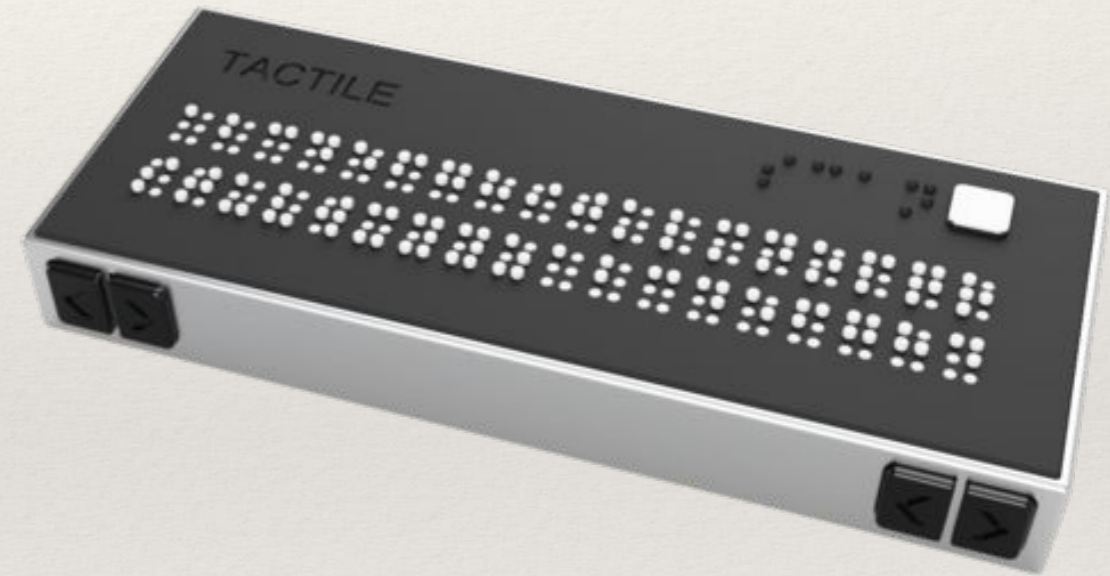
Pressing a button, displaying text, playing sound are all interactions that are describable semantically.



---

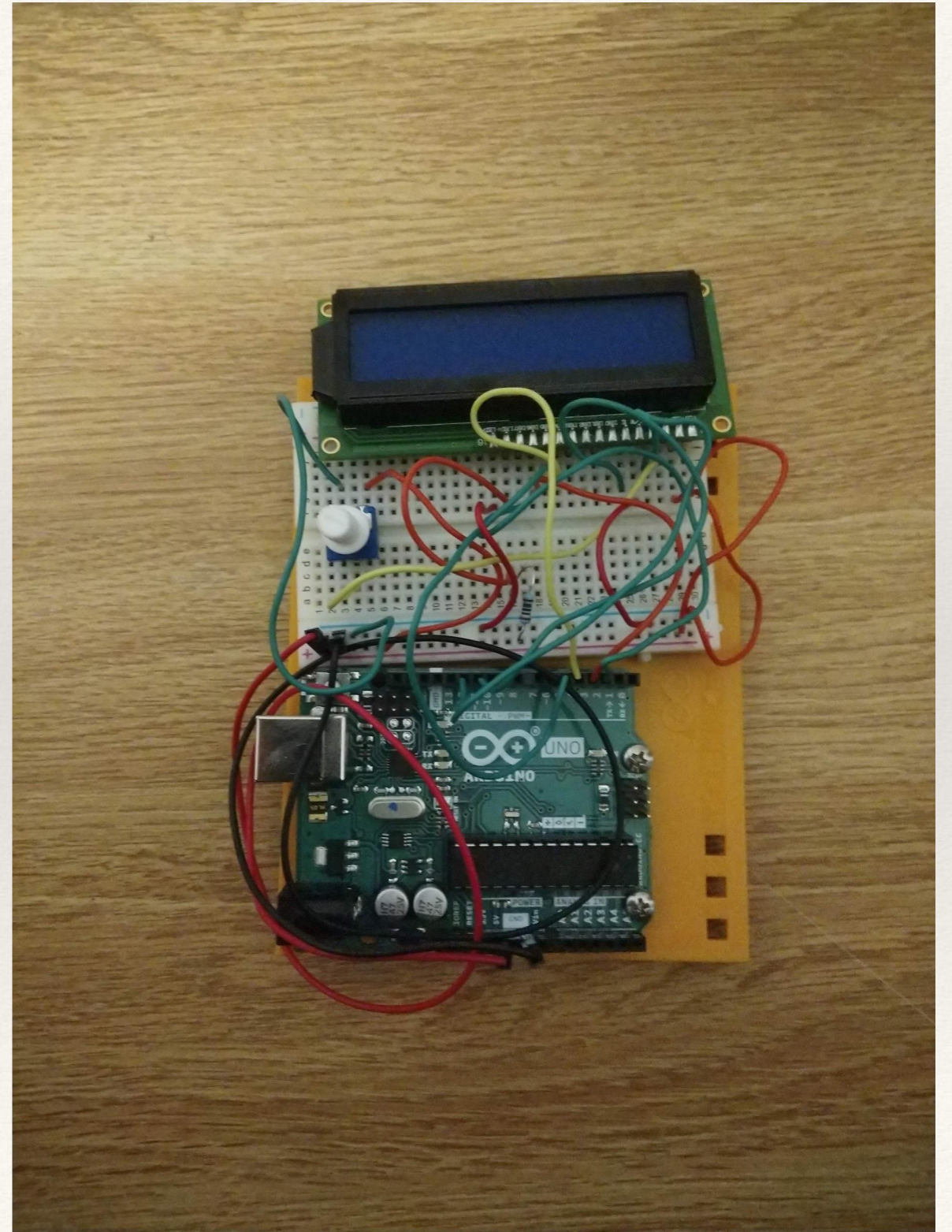
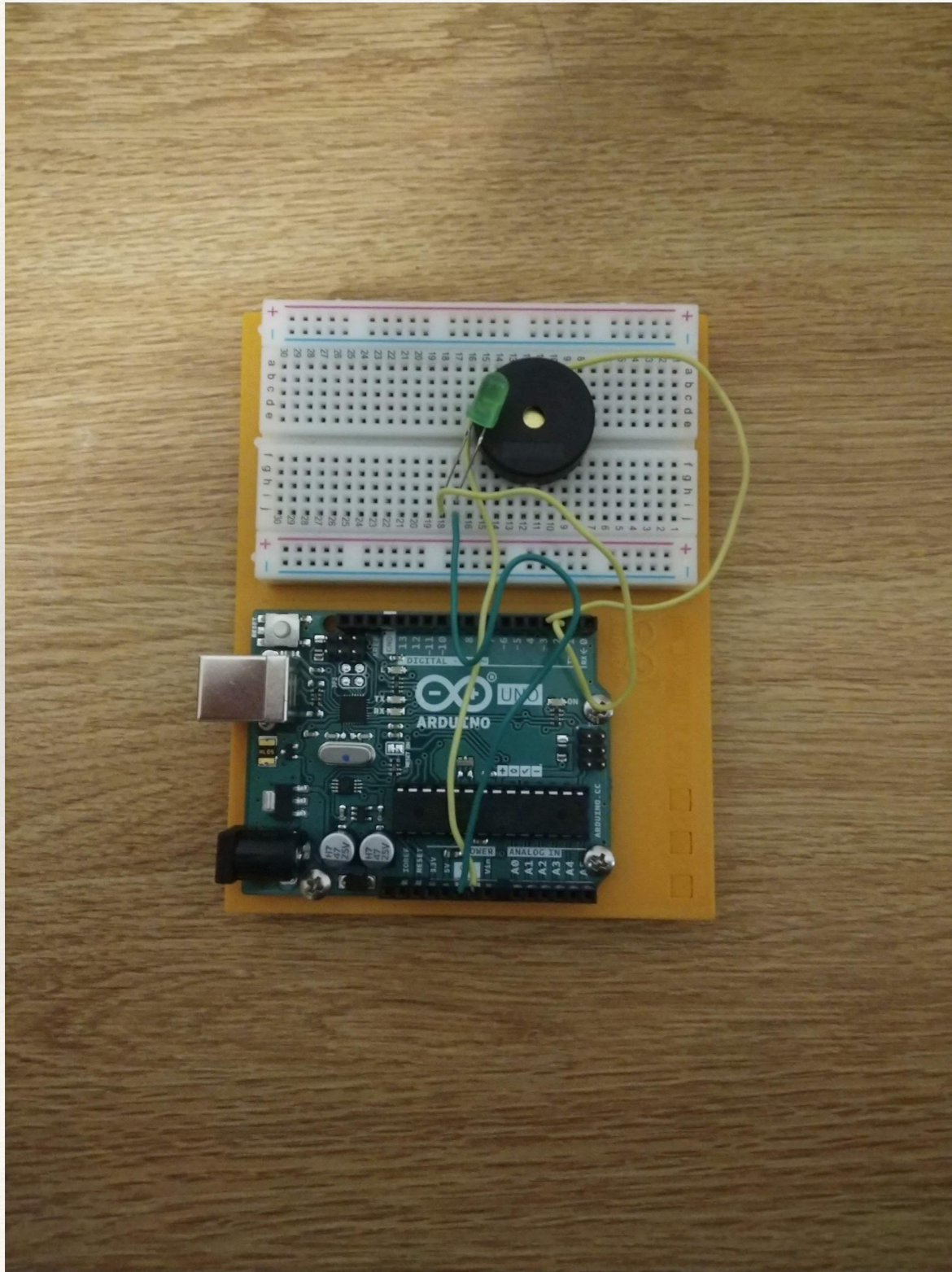
# Hello World!

---





# Demo





---

# What is SIOP?

---

Semantic Input/Output Protocol: a hypothetical protocol that describes I/O semantically allowing an universal abstraction between hardware and software

Devices of different types can effectively work together and interactions would be optimized for the virtues of a particular device being used.





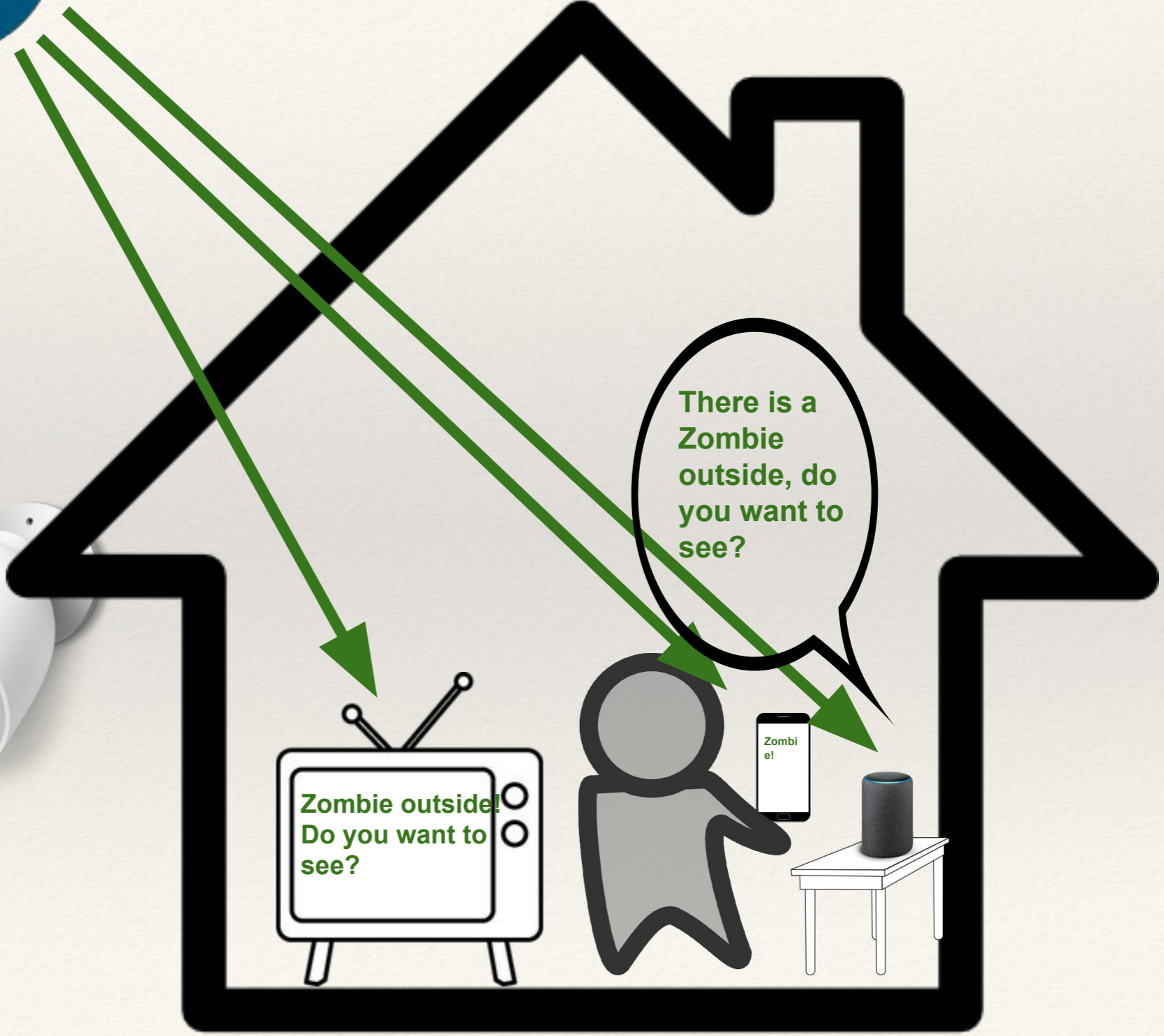
Video Streamed to cloud service





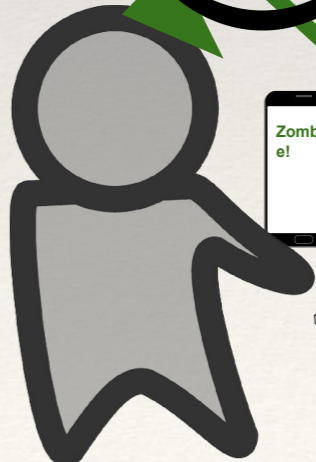


Cloud service AI algorithm discovers zombie outside, sends messages to smartphone, smartspeaker and tv.

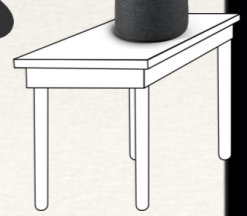


There is a  
Zombie  
outside, do  
you want to  
see?

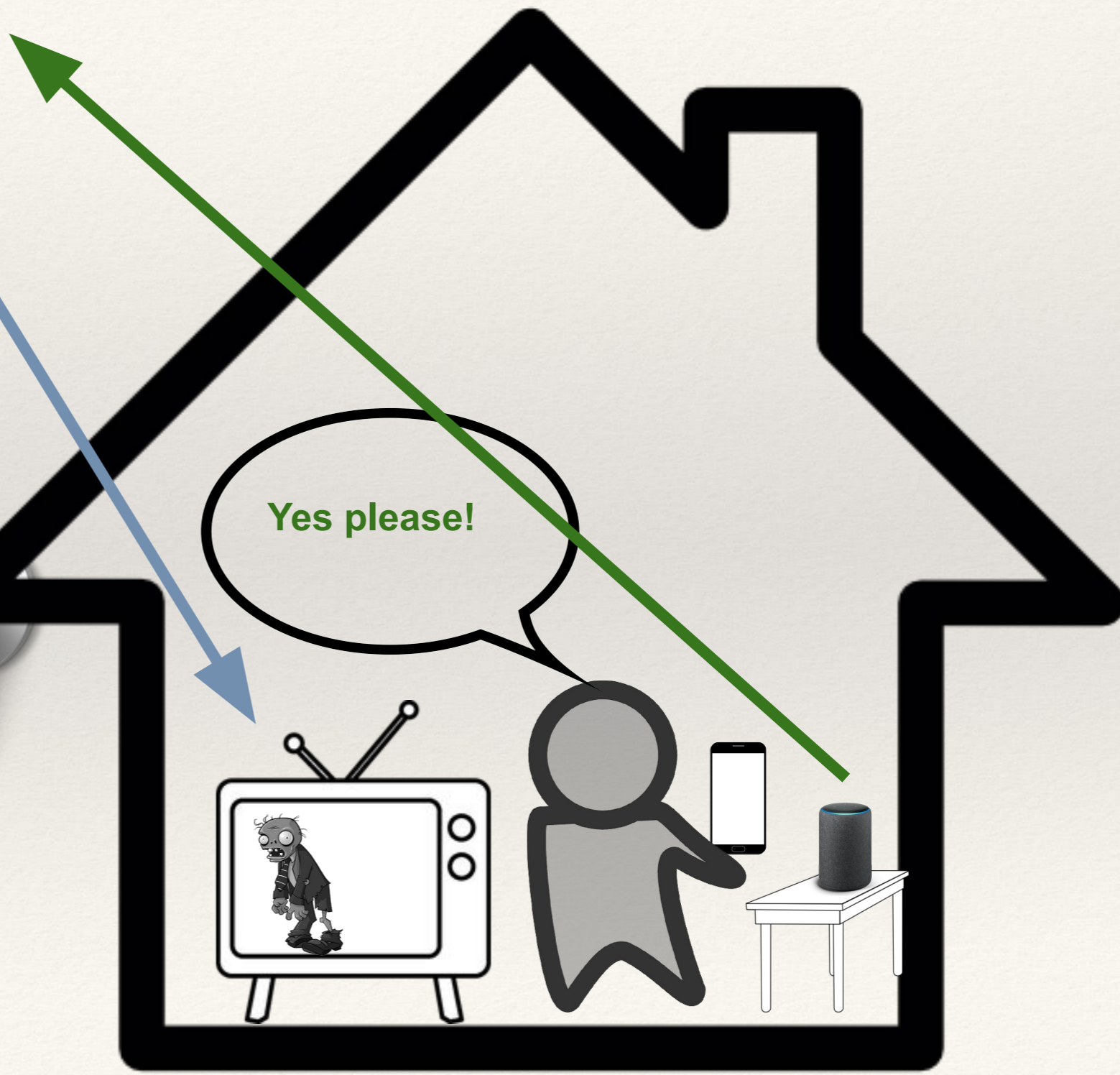
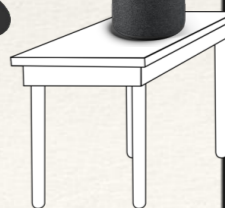
Zombie outside!  
Do you want  
to see?



Zombi  
e!









---

# SIOP Client

---

Software services are generally cloud based.

Services have interaction points for service I/O.

- Users associate their hardware and their software interaction points.
- Hardware has a manifest file identifying types of interactions it's hardware supports. Services use these manifests to identify hardware that can be linked to interaction points



# Hardware / Software Roles

Hardware:	Software:
<ul style="list-style-type: none"><li>● Run SIOP Client</li><li>● Contains manifest file describing which interactions are supported by hardware</li><li>● Contains interaction lookup table in order to notify services “listening” for interactions with device</li><li>● Hardware specific implementation of interaction responsibility of the SIOP client.</li></ul>	<ul style="list-style-type: none"><li>● Service, not application oriented<ul style="list-style-type: none"><li>○ In its purest form, the SIOP paradigm implies that software would not need to concern itself with UI and interaction management.</li></ul></li><li>● Cloud based<ul style="list-style-type: none"><li>○ Some device caching might be possible</li></ul></li><li>● Interfaces = “Interaction Points”</li><li>● Users might purchase, manage access and licences in an “online marketplace” similar to an app store.</li></ul>



---

# Central Principles

---

- Cloud based
- Semantic I/O
- Software Centric
- Service Focused
- Loosely Coupled



---

# Cloud Based

---

Hardware's primary responsibility is interaction implementation.

Services primary responsibility is processing

Because semantic I/O is the primary abstraction, other abstractions such as Operating Systems are not required

Some offline caching would likely be possible, but would become an increasing counter to the core paradigm as amounts and complexity increased.



---

# Semantic I/O

---

**Problem #1** (the blocking problem): How to define interactions semantically?

**Ontologies** define the formal semantics of the terms used for describing data and the relationships between these terms. They ensure that meaning of data exchanged between and within systems is consistent and shared both by humans and computers



---

# Work Accomplished

---

This Spring **David Jennings**, a student working with me on his Honors Thesis, wrote “An Ontology For Semantic I/O” on the subject.

*“Think about how the users interact with their devices, what the user hopes to achieve through this/these interaction(s), and come up with how to structure messages that would allow these devices to interact with each other. “*



---

# Current State

---

I have built on this research and have laid out the following starting point for describing I/O semantically.

## 5 Categories of I/O (Content Type)

1. Text
2. Primitives
3. Gesture / Movement
4. Audio
5. Visual

There is also a “Delivery Type” which is a sub-category.

- Ex: Text/Eng-US



# Example Message

SIOP v1.0 INTERACTION.INPUT

messageId: 3e88c44e-1693-11e9-ab14-d663bd873d93

originId: aa205739-03ce-4431-99ea-1f2517735d48

originUri: siop://doorbell.home.gormanly.com

destinationId: 2e39eae6-1693-11e9-ab14-d663bd873d93

destinationUri: siop://myHomeSecurityApp.com

message time stamp: 1547315953507

message context: 0

Content Type: Button Press : Boolean

Value: true

Fig. 1. Example SIOP message



---

# Software Centric

---

Paradigm promotes a software centric approach.

- Hardware is paired to interactions where it can carry out intent of the software.
- Software is no longer engineered to “run on” a particular hardware platform (with the exception of the web server running the services)



---

# Service Focused

---

Software functionality focuses on services provided, not interface.

- Software loses its “Application” components
- UI is no longer a service concern, intent is carried out by the individual pieces of Hardware



---

# Loosely Coupled

---

Software services and hardware clients need to be connected to allow interactions

- Some concept a marketplace on central management software allows users to connect available hardware and software via compatible interactions
- Users can switch out software and hardware connections at any time



---

# Inspiration

---

This approach has been heavily influenced by the ideas and work behind the **semantic web**.

- Berners-Lee, Tim, James Hendler, and Ora Lassila. "The semantic web." *Scientific american* 284.5 (2001): 28-37.
- Shadbolt, Nigel, Tim Berners-Lee, and Wendy Hall. "The semantic web revisited." *IEEE intelligent systems* 21.3 (2006): 96-101.
- Vrandečić, Denny. "Ontology evaluation." *Handbook on ontologies*. Springer, Berlin, Heidelberg, 2009. 293-313.



# Questions?

Brian Gormanly

<http://linkedin.com/in/gormanly>