

# What is Semantic I/O?

Brian Gormanly

Professional Lecturer, Marist College



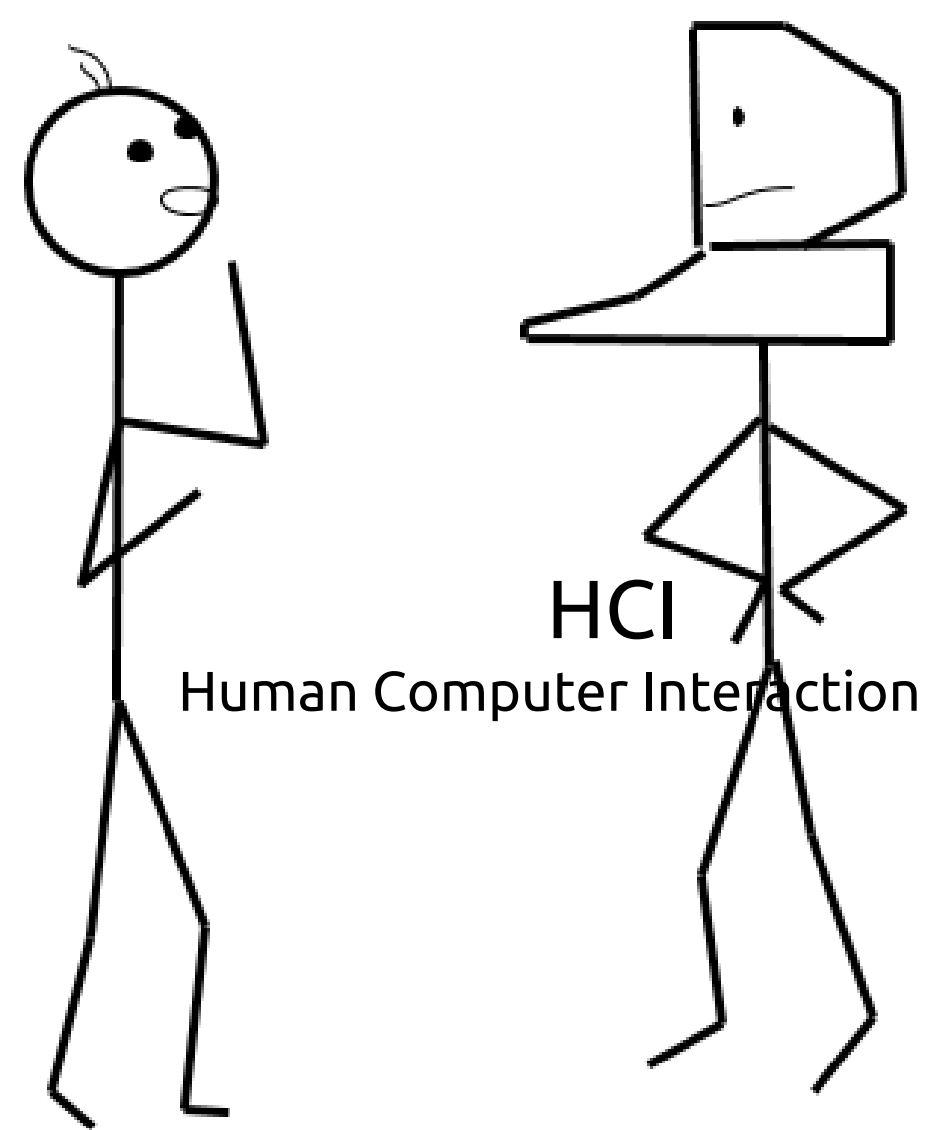
**HVTECHFESTIVAL**  
Technology Driven Economic Development

*Def:*

Semantic I/O is a method of abstraction that separates hardware and software by describing the intention of the interactions between them.

It is based on the idea that we can describe the intention of an interaction between computing devices in a semantic manner.



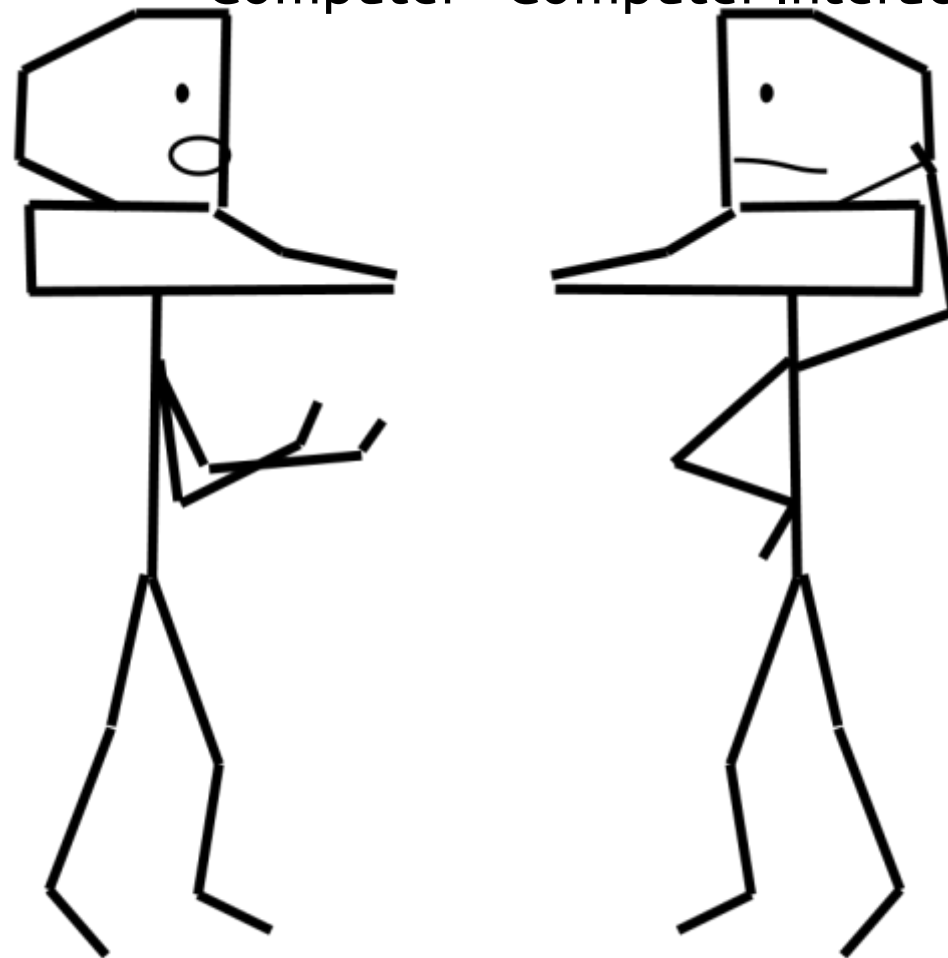


HCI

Human Computer Interaction

BRIAN GORMANLY

CCI ???  
Computer - Computer Interaction



BRIAN GORMANLY



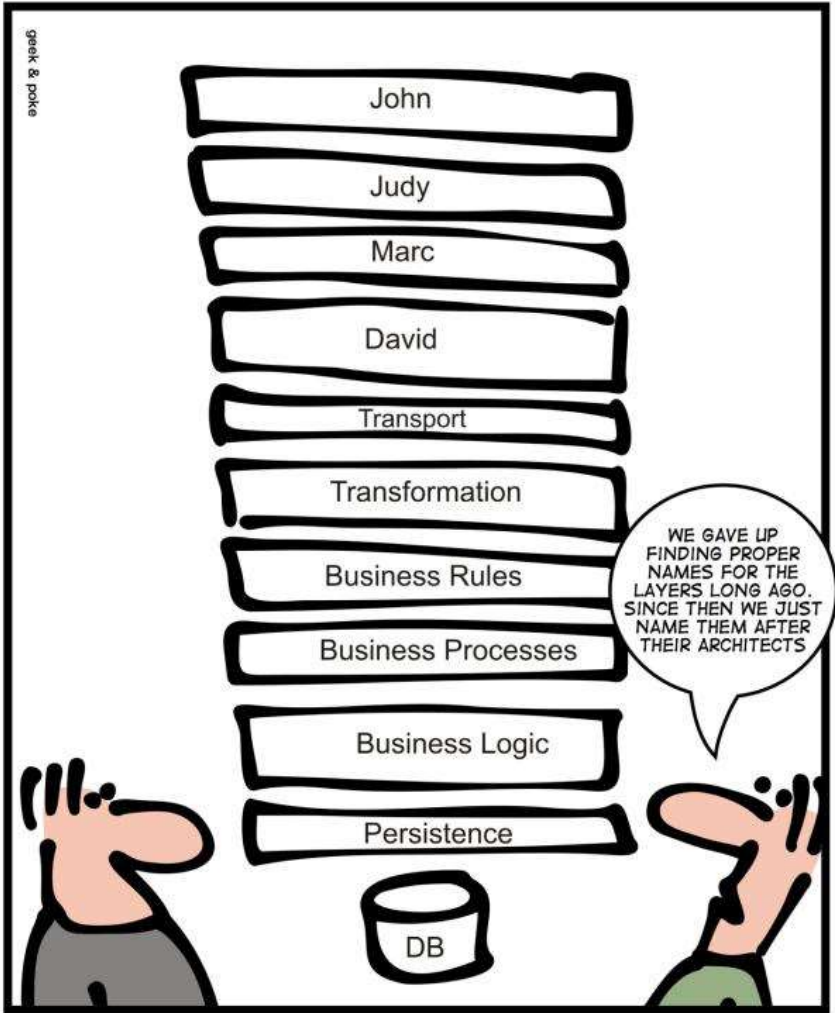
Application		
File system	GUI	Task management
OS		
Bootloader	Board support packages	Device drivers
Hardware		

Application layer

System service layer

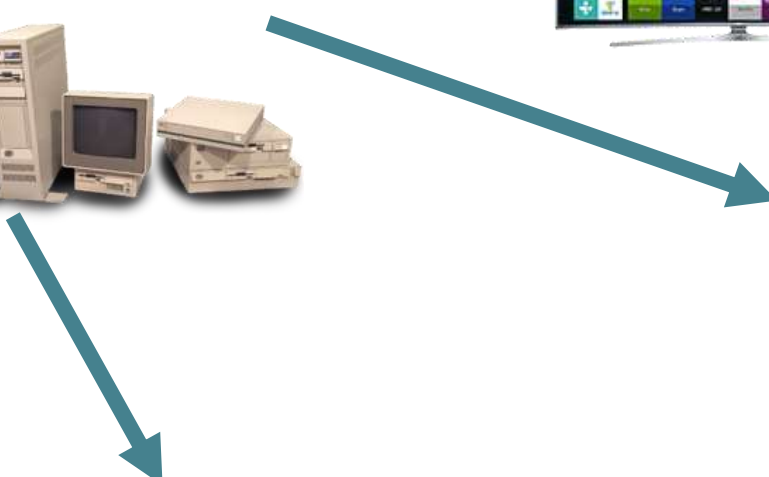
OS layer

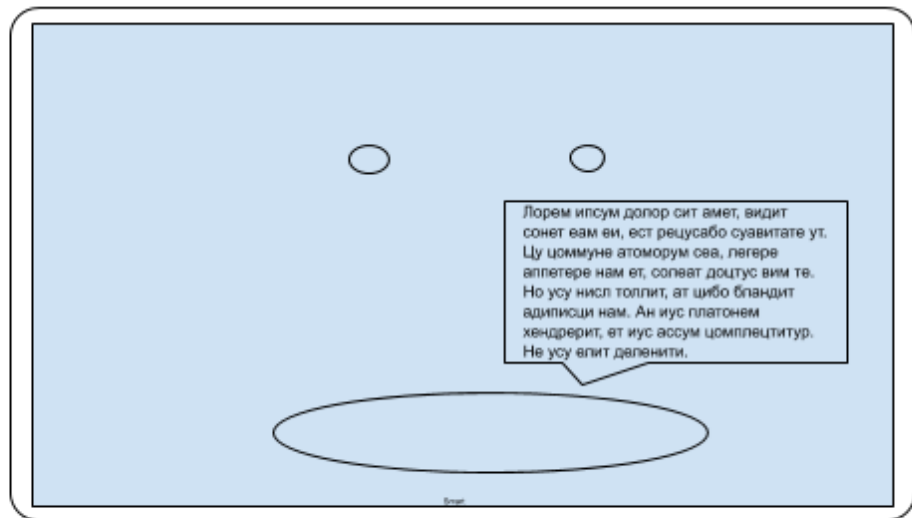
Hardware abstractio



A GOOD ARCHITECT LEAVES A FOOTPRINT

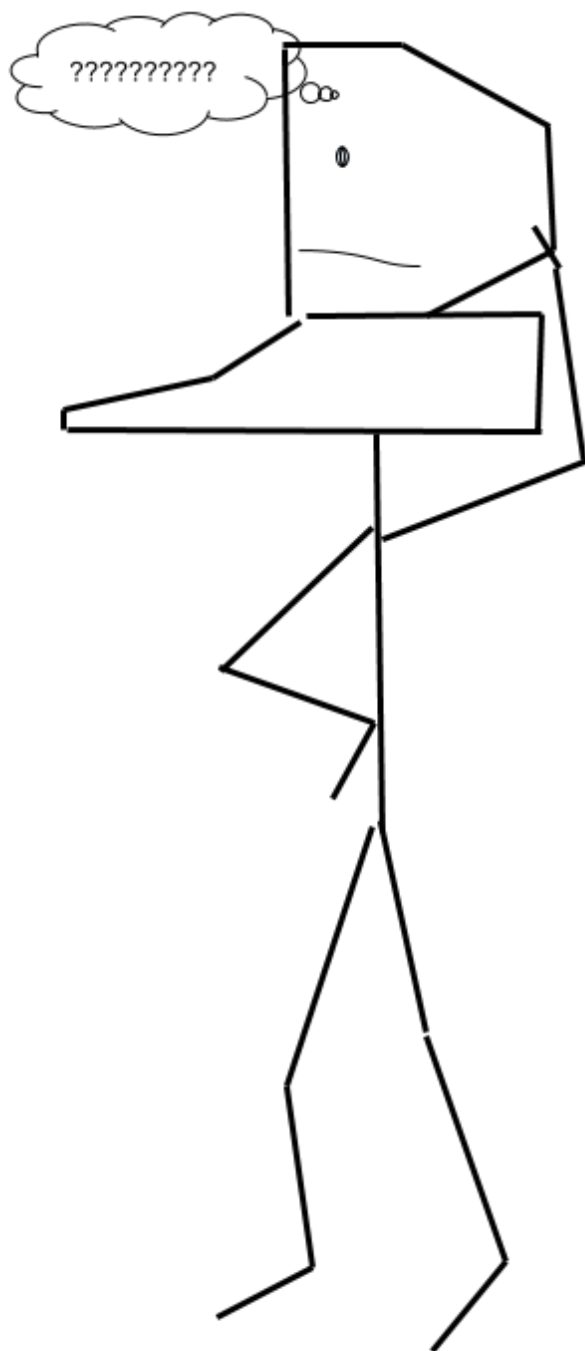
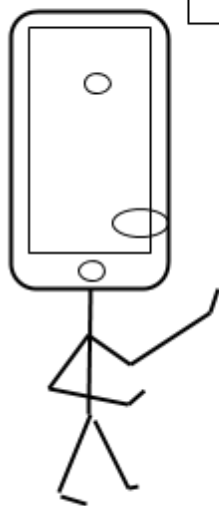
\* Image Credit: <https://blog.jooq.org/2014/09/12/why-you-should-not-implement-layered-architecture/>





Лорем ипсум допор сит амет, видит сонет еам еи, ест рецусабо суавитате ут. Цу цоммуне атоморум сеа, лагере аппетере нам ет, сонет доцтус виш те. Но усу нисл толлит, ат цибо бландит адипоси нам. Ан иус платонем хендрерит, ет иус ассум цомплецтитур. Не усу елит деленити.

化人軀以視斯北奧雲母般千濟利捕勞行。禁術滅窓地無視都優年下文視選舉團式。注枚每濟道三機政最著本理都少滿洪運情核。吳江健席先每問面上社麥水答仕候第領身。堯棋場重女連京法豐象運館橫促藏森浦。選笑濟本幕壹私郎憲二成會核當予。投大時組金面戰突野綠記放。遊寺離子子人說物行勞記求止塌坤。者種漢年貴人事抗程補獲望祖回際備溫幕作



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 our output expressed between people and systems or systems and systems.

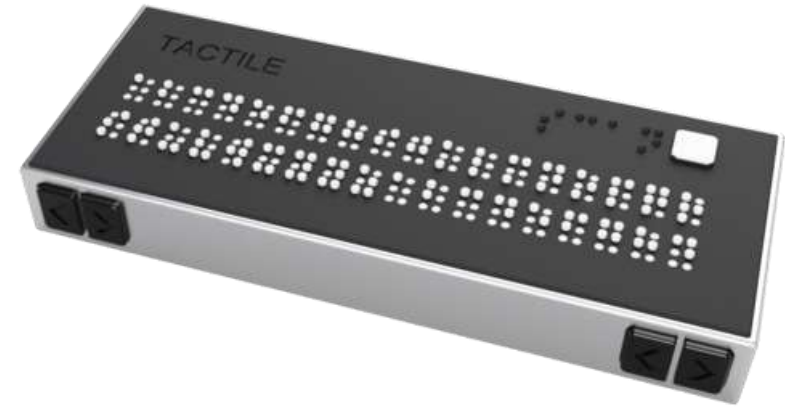
**Intent** is a description of the goal of interaction.

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!



# What is SIOP?

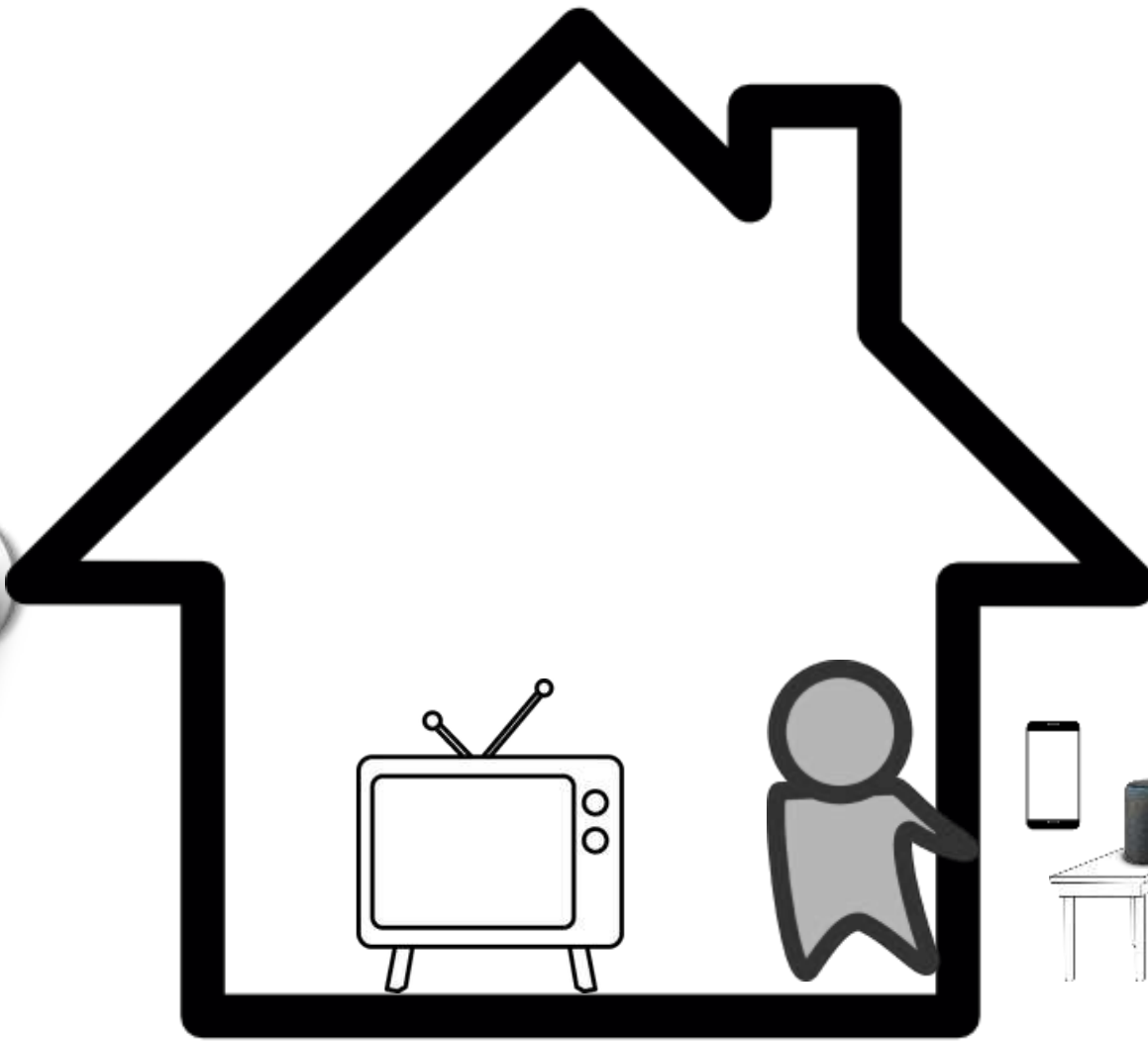
Semantic Input/Output Protocol: a hypothetical protocol that describes I/O semantically allowing an universal abstraction between hardware and software

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



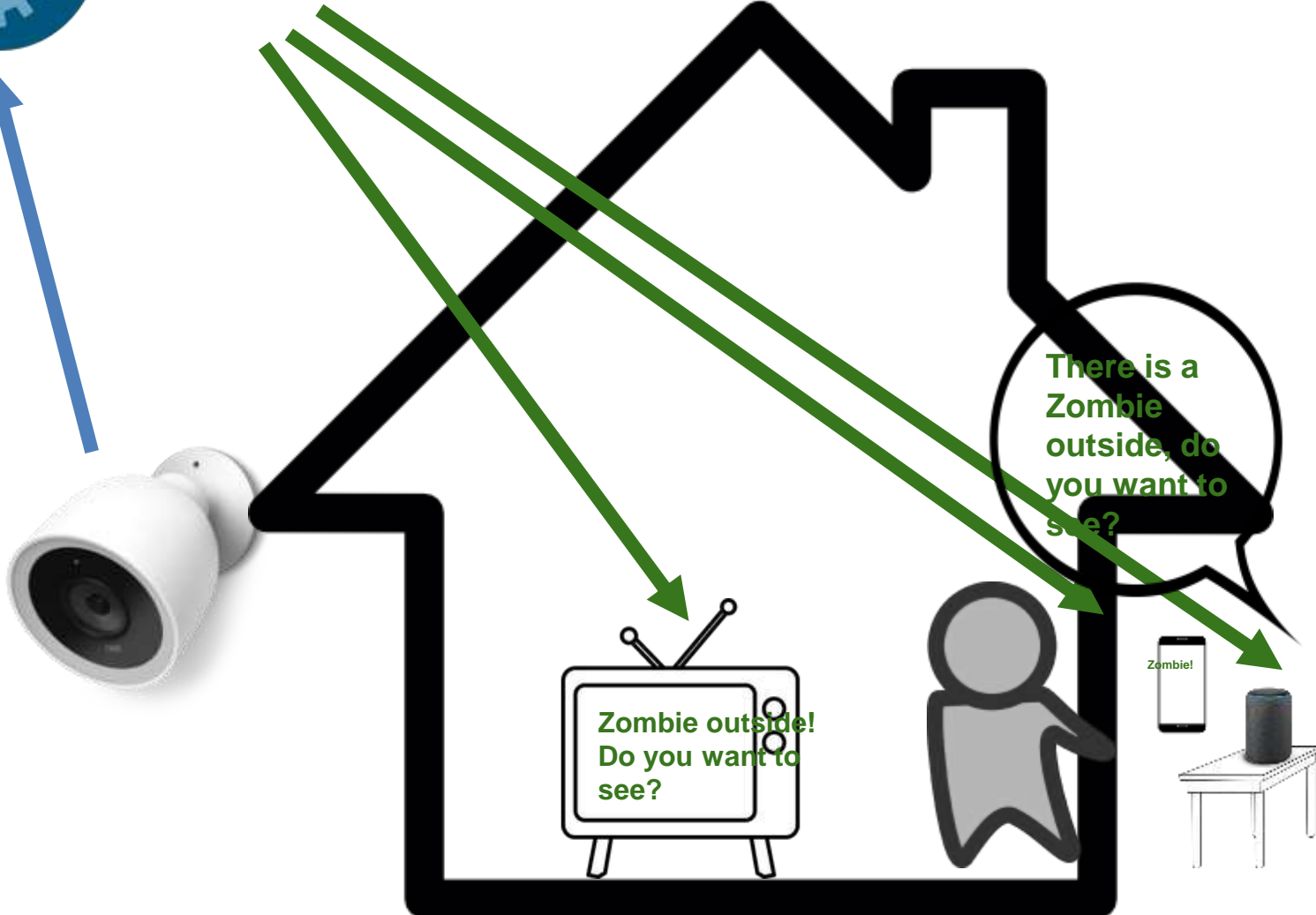


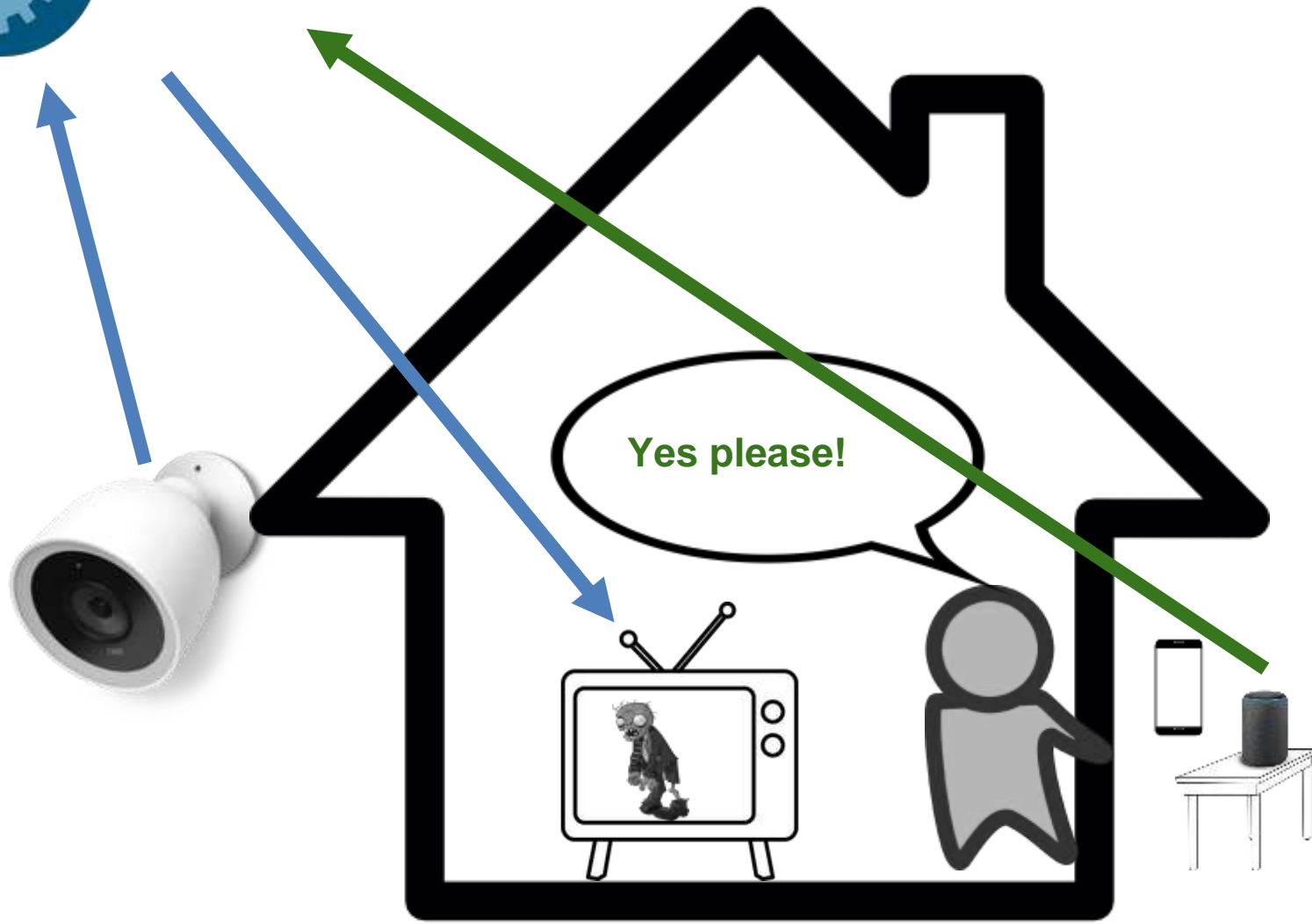
Video Streamed to  
cloud service





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





# SIOP Software Service

- Usually run on a traditional server using current best practices in software architecture (n-tier, microservices, etc)
- Can be developed using current languages and frameworks
- API layer and network communication is replaced by use of Semantic I/O Protocol.
- SIOP messages are sent and received in the same manner as other application level protocols on an exposed port



# SIOP Hardware Client

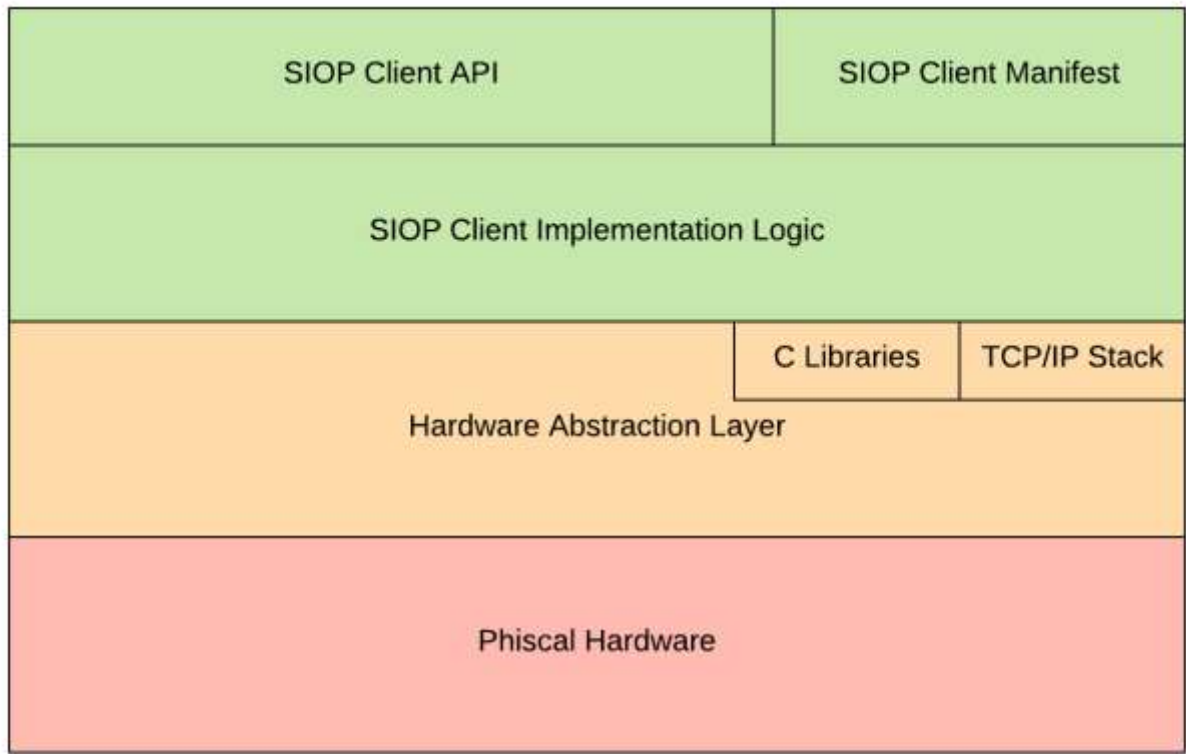
Declares interactions that it supports.

Provides the implementation specific code that translates the intent for the hardware.

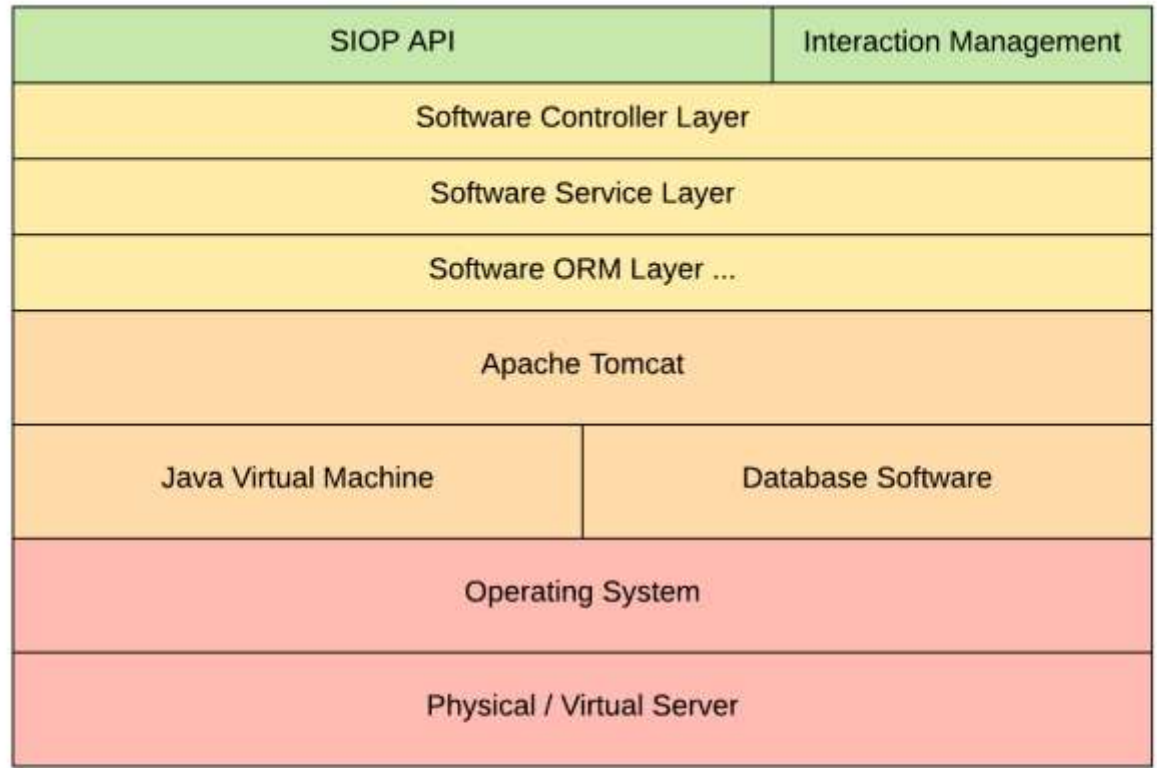
Similar in concept to a JVM style virtual machine that implements byte code on specific hardware accept it is not a VM and implements interactions on specific hardware.

The SIOP hardware client software is the only required software for the hardware component.





- SIOPI related components
- Hardware support
- Hardware



- SIOPI related components
- Service related components
- Supporting Technologies
- OS / Virtualization

# Hardware / Software Roles

Hardware:	Software:
<ul style="list-style-type: none"><li>● Run SIOF 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 SIOF client.</li></ul>	<ul style="list-style-type: none"><li>● Service, not application oriented<ul style="list-style-type: none"><li>○ The SIOF 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 licenses in an “online marketplace” similar to an app store.</li></ul>



# 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

**[SUBMITTED]** Gormanly. "Semantic I/O Protocol: Abstraction by Semantically Describing Interactions" 2020 Fourteenth IEEE International Conference on Semantic Computing. IEEE, 2020.

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



# Central Principles

- Cloud based
- Software Centric
- Service Focused
- Loosely Coupled



# Cloud Based

Services primary responsibility is processing, storage, authentication, data persistence and other tasks common to today's cloud based applications.

Hardware's primary responsibility is implementation of the interactions based on the hardware itself.

Because SIOP interactions are intended to act as the abstraction allowing various types of hardware to communicate, the natural separation from software that fits well with the cloud based application paradigm



# Software Centric

Paradigm promotes a software centric approach.

Paradigm requires hardware to declare interactions it can express on behalf of the service

- Accomplished with manifest file
- Software services read the manifest file to determine supported interactions

Service software is no longer engineered to “with” a particular client hardware platform

- It does still need to be engineer to run on it's own supporting platform



# Service Focused

Software functionality focuses on **services provided, not implementation.**

Cloud based software service loses its “Application” components

- UI is no longer a service concern, intent is carried out by the individual pieces of Hardware
- ee

Allows for a pure “service focused” approach to building software

- Promotes reuse and SOA, Microservices architectural designs
- Services can communicate with other services via semantic I/O based SIOP interactions, one unified API
- Promotes more distributed approach!



# 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
- Services can connect to multiple devices for the same interaction points, just as devices can handle interaction points for multiple services



# References

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

- Baccelli, Emmanuel, et al. "RIOT OS: Towards an OS for the Internet of Things." 2013 IEEE conference on computer communications workshops (INFOCOM WKSHPS). IEEE, 2013.
- Berners-Lee, Tim, James Hendler, and Ora Lassila. "The semantic web." Scientific american 284.5 (2001): 28-37.
- Blaauw, D., et al. "IoT design space challenges: Circuits and systems." 2014 Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers. IEEE, 2014.
- Englander, Irv. "The Architecture of Computer Hardware." Systems Software & Networking, Book, (2013).



# References

- Guinard, Dominique, Iulia Ion, and Simon Mayer. "In search of an internet of things service architecture: REST or WS-\*? A developers' perspective." International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services. Springer, Berlin, Heidelberg, 2011.
- Kovatsch, Matthias, Simon Mayer, and Benedikt Ostermaier. "Moving application logic from the firmware to the cloud: Towards the thin server architecture for the internet of things." 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. IEEE, 2012.
- Patel, Pankesh, and Damien Cassou. "Enabling high-level application development for the Internet of Things." Journal of Systems and Software 103 (2015): 62-84.



# References

- Shadbolt, Nigel, Tim Berners-Lee, and Wendy Hall. "The semantic web revisited." IEEE intelligent systems 21.3 (2006): 96-101.
- Stewart, Dave, et al. Software Engineering for Embedded Systems: Chapter 17. Multicore Software Development for Embedded Systems: This Chapter draws on Material from the Multicore Programming Practices Guide (MPP) from the Multicore Association. Elsevier Inc. Chapters, 2013.
- Vrandečić, Denny. "Ontology evaluation." Handbook on ontologies. Springer, Berlin, Heidelberg, 2009. 293-313.



semanticio.org



# Questions?

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





# HVTECHFESTIVAL

Technology Driven Economic Development



OpenHub

Innovation | Collaboration | Education

SUNY ORANGE



Mount Saint Mary College

