

Smart Home Infrastruktur

Ein Framework für eine zeitgemäße Umsetzung

Alexander Sowitzki

Hochschule für Angewandte Wissenschaften Hamburg

17. April 2018

Historie

- Smart Home ist frischer Begriff
- Viele unterschiedliche tlw. proprietäre Lösungen

Probleme

- Komplexe Anwendungsszenarien
- Verteiltes System
- Aktualität kritisch
- Kompatibilität
- Latenzzeiten, Durchsatz & QoS
- Heterogene Datenarten

Fragestellung

Wie entwickelt man ein Smart Home, welches zeitgemäßen Anforderungen genügt?

Testfeld Puppenhaus

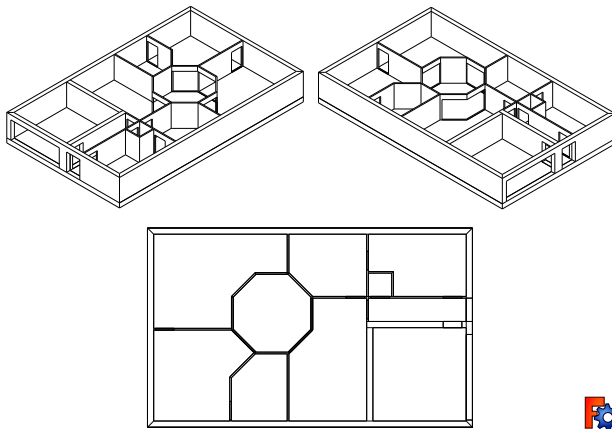


Figure: Hausentwurf[1]

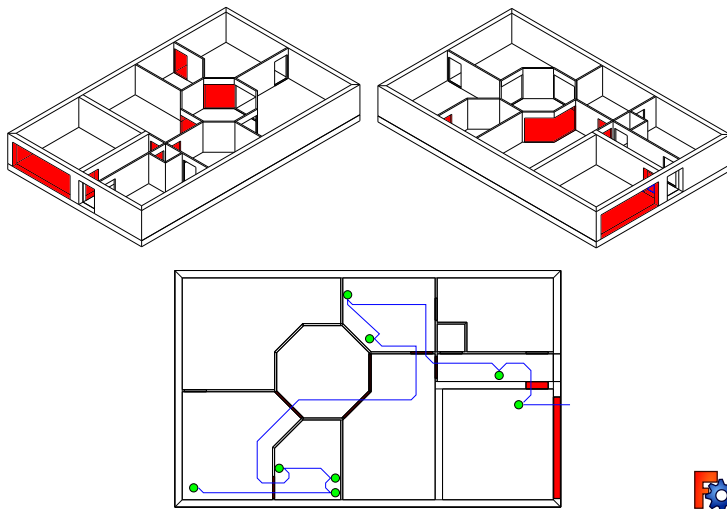


Figure: Aufstehen[1]

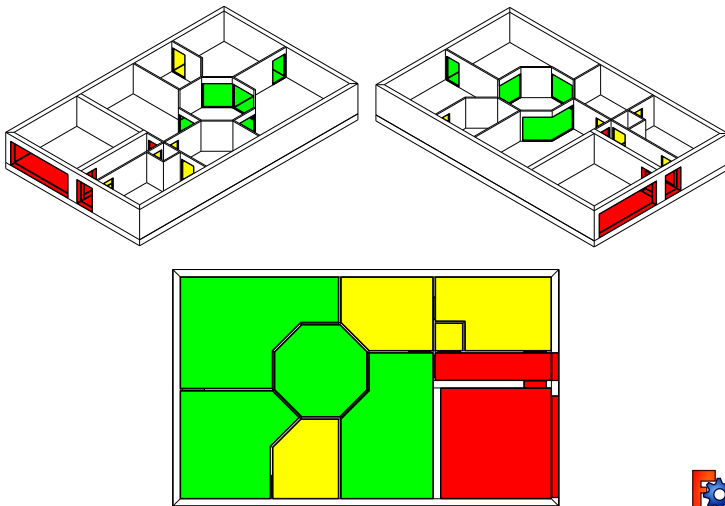


Figure: Bird Alert[1]



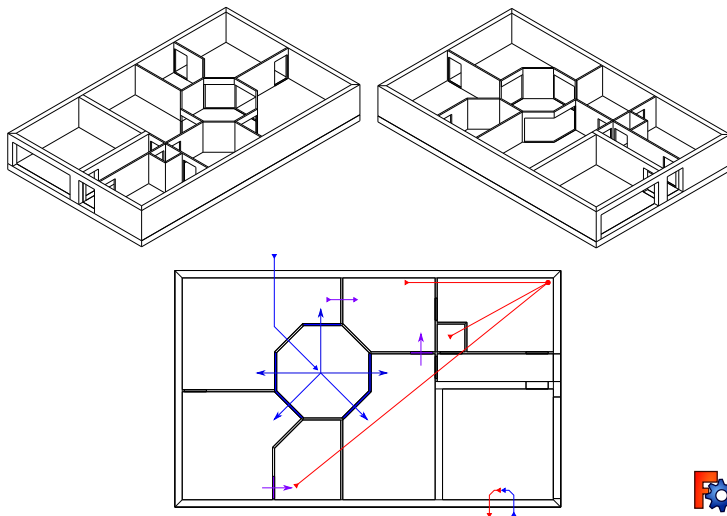


Figure: Belüftungsregelung[1]

Besonderheiten Puppenhaus

- Einfache Verkabelung
- Hohe Übersicht
- Hohe Kontrolle

Wichtige Gemeinsamkeiten

- Verteiltheit
- Komplexe Datenaggregationen
- Zentrale Konfiguration

Vereinfache Aspekte der Realität

- Chaotisches Verhalten von Menschen
- Anpassbare Räumlichkeiten
- Privatsphäre

Software Experimente

- Verteilte Entscheidungsfindung
- Übertragung von Kameradaten und Beleuchtungsinformationen
- Föderationssysteme
- Datenhaltung und deren Struktur im Pubsub-System
- Assistenz bei Treiberentwicklung

Smart Home als Multiagentensystem

- Persistente Livekonfiguration aller Agenten über Pubsub
- Logausgaben können geteilt und gesammelt werden
- Präsenz und Status von Agenten einsehbar
- Selbstauskunft & automatische Serialisierung in Binärformat
- Ströme über Out-Of-Band Kommunikation

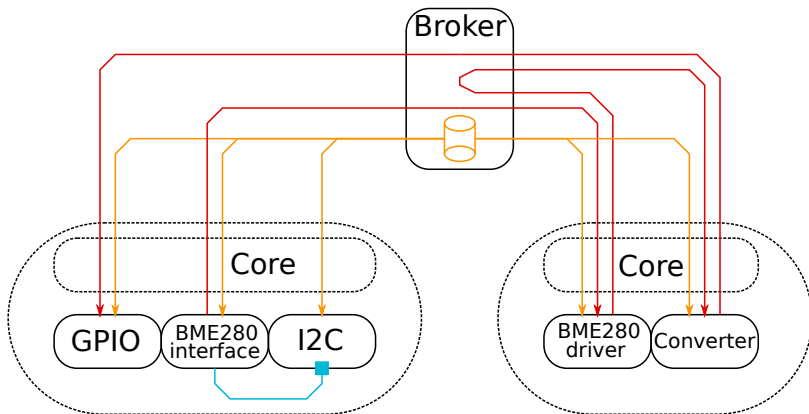


Figure: Netzwerkvisualisierung[1]

Docker für Datenverarbeitung

- Thema des Grundprojekts
- Nicht auf Ressourcen angewiesene Agenten als Docker Container
- Verteilung in Systemen wie Docker Swarm und Kubernetes
- Deckt Ausfallsicherheit und Lastverteilung ab
- Automatisches Bauen und Verteilen über Open Source Systeme

openHab / Eclipse Smart Home

- Sieht sich als Verbindungsglied zwischen Plattformen
- Systeme werden per REST angebunden
- Zentralisierte Datenverarbeitung ohne HA
- Ähnliches für FHEM und NodeRed [2][3]

OCF - IoTivity

- OCF spezifiziert Kommunikation und versucht sich als Standard zu etablieren
- IoTivity ist unabhängige Implementation
- Schwerpunkt auf CoAP [4]
- Schwergewichtige Entwicklungsumgebung
- Keine Unterstützung von Offlinekonfiguration

Abgrenzung[6]

- Zielgruppe sind fortgeschrittene Anwender
- Entwicklerzeit
- Anbindung an Betriebssystem
- Bandbreiten & Prozessoreffizienz durch optimiertes Protokoll [5]
- Erweiterbarkeit & Integrationsfähigkeit

Ausblick

- Modell konzipiert
- Bau steht bevor
- Arbeit am Framework begonnen

Bibliographie I



Alexander Sowitzki. *Eigenarbeit*. 2018.



FHEM. URL: <https://fhem.de> (visited on 04/10/2018).



Node-RED. URL: <https://nodered.org> (visited on 04/10/2018).



IoTivity. URL: <https://www.iotivity.org> (visited on 04/10/2018).



HTTP vs MQTT performance tests. URL: <https://flespi.com/blog/http-vs-mqtt-performance-tests> (visited on 04/10/2018).



Erik Nimmermark and Alexander Larsson. “Comparison of IoT frameworks for the smart home”. In: (2016).