Architecting Internet of Things based Agri-Food Systems

Cagatay Catal, Bedir Tekinerdogan
Wageningen University & Research (WUR)
Information Technology Group (INF)
Wageningen, The Netherlands
cagatay.catal@wur.nl
bedir.tekinerdogan@wur.nl

WAGENINGEN UR
For quality of life

Banja Luka, January 23, 2020

IBM Mainframe 370, year 1964

Apollo 11 Command Module **(1965)**
64 kilobytes of memory
operated at **0.043MHz**

An **iPhone 11**, CPU

Hexa-core (2× high power cores at **2.66 GHz** + 4× low power cores at 1.82 GHz)

has **4GB of memory**

Computing Power

- Smaller size
- More powerful
  - More memory and more storage
- Cheaper

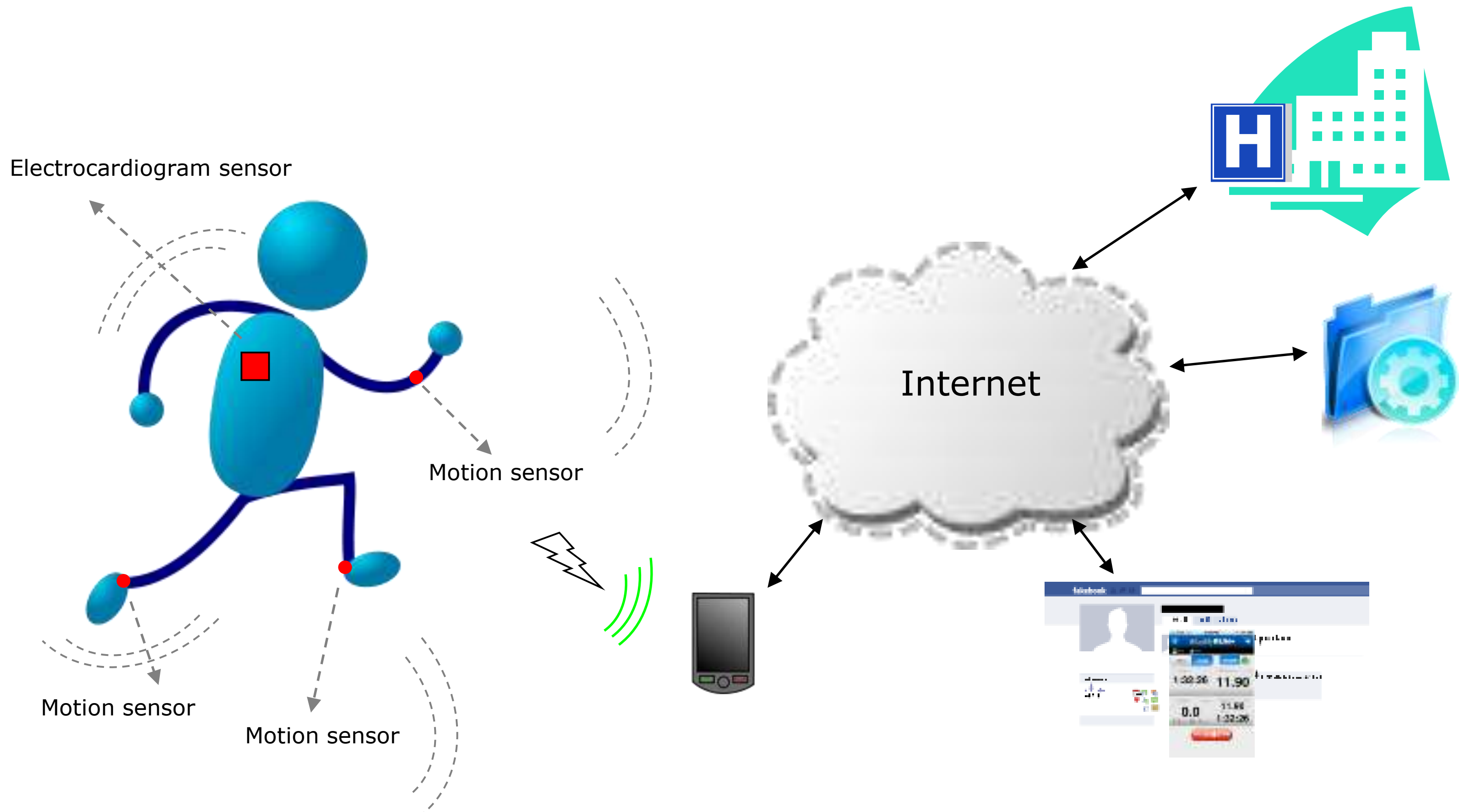
Sensor Devices Becoming Widely Available...

- Programmable devices
- Off-the-shelf tools

More “Things” are being connected

Home/daily-life devices
Business and
Public infrastructure
Health-care
…

People Connecting to Things



Things Connecting to Things

What is a Thing ?

- An object of the **physical world** (physical things)
  or the **information world** (virtual things)
- capable of being **identified** and **integrated** into communication networks

(ITU, 2015)

Things ?

- We can turn almost **every object** into a "thing"
- A "thing" looks like an **embedded system** currently
- A "thing" generally consists of **four main parts**:
  - **Sensors & actuators**
  - **Microcontroller**
  - **Communication unit**
  - **Power supply**

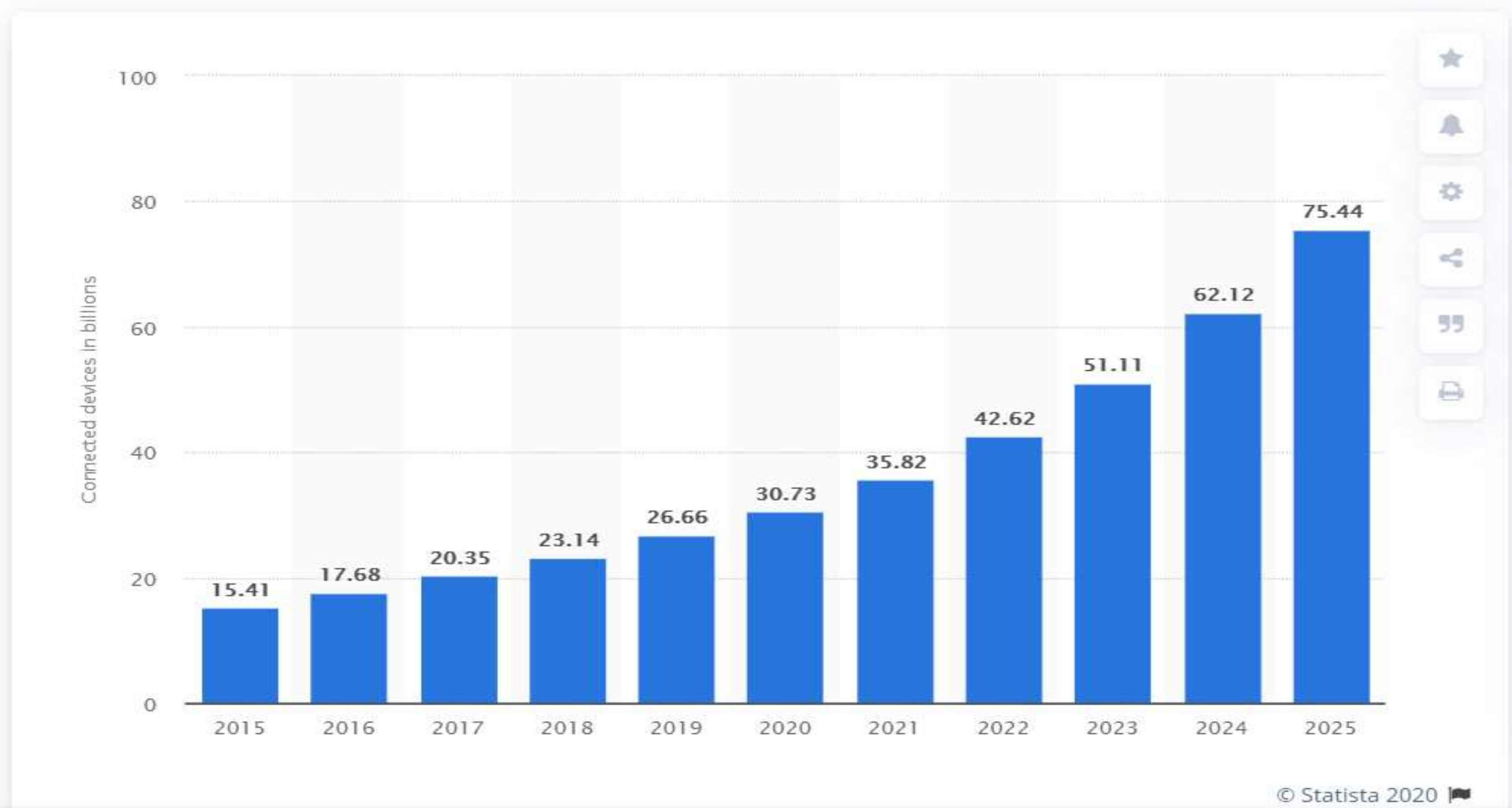
Sensors & Actuators

- **Sensors**
  - They are mainly input components
  - They sense and collect surrounding information

- **Actuators**
  - They are mainly output components
  - They alter the surrounding
  - Examples:
    - Adding lighting, heat, sound
    - Controlling motors to move objects
    - Displaying messages

Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025

(in billions)



5-fold increase in 10 years !

https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

Internet of Things (IoT)

- Internet of Things (definition by ITU)
  - **Network of objects** or ‘things’
  - embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data

IoT

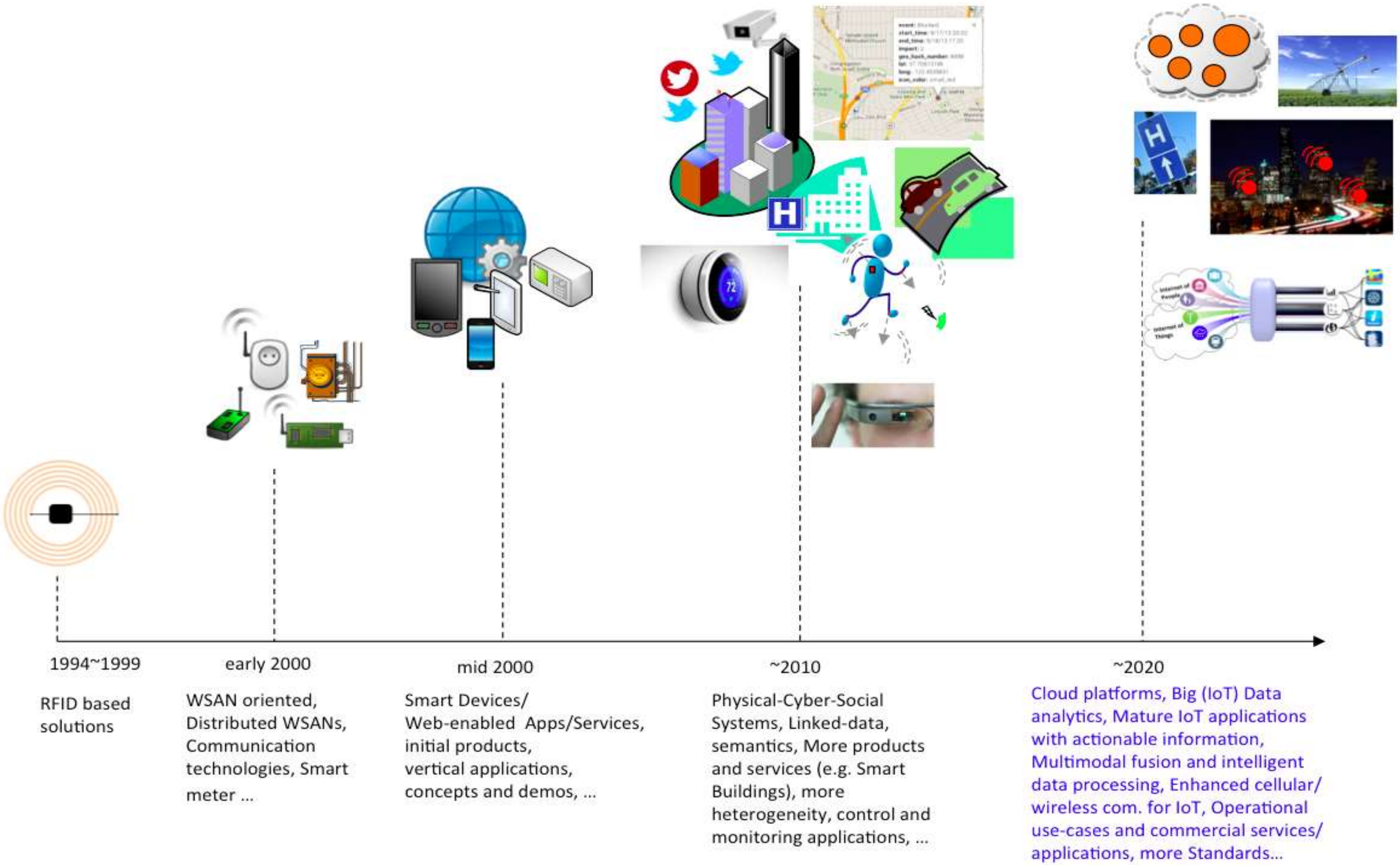
- IoT = **Physical Object** +

  Controller, Sensor, and Actuators +

  Internet

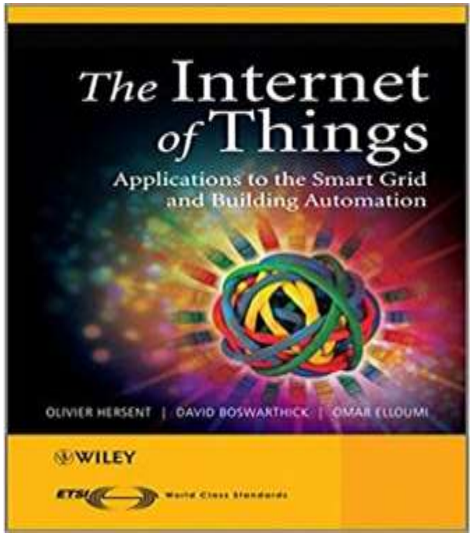
(McEwen and Cassimally, 2014)

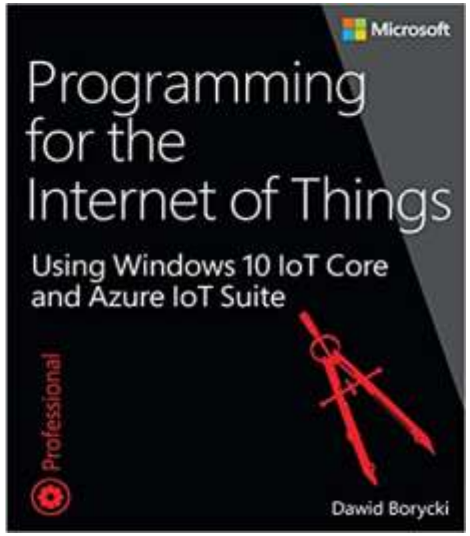
History of Internet of Things

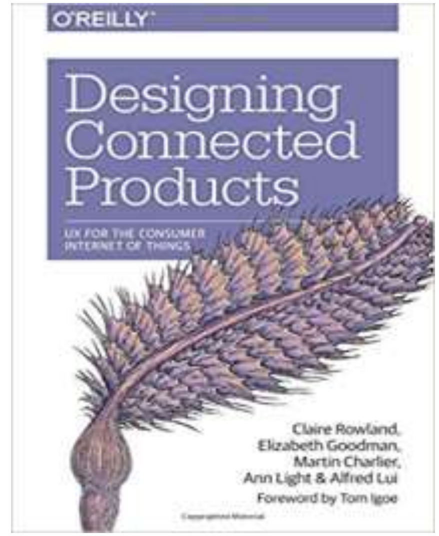


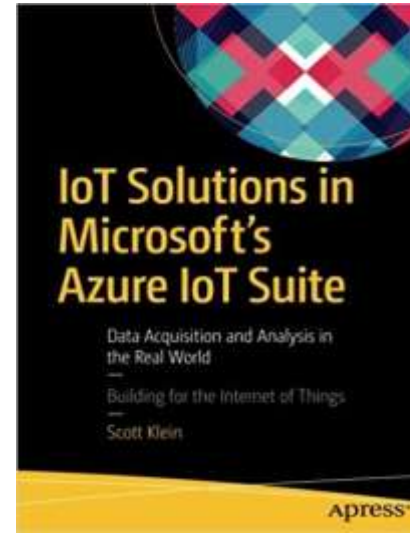
P. Barnaghi, A. Sheth, "Internet of Things, The story so far", IEEE IoT Newsletter, September 2014.

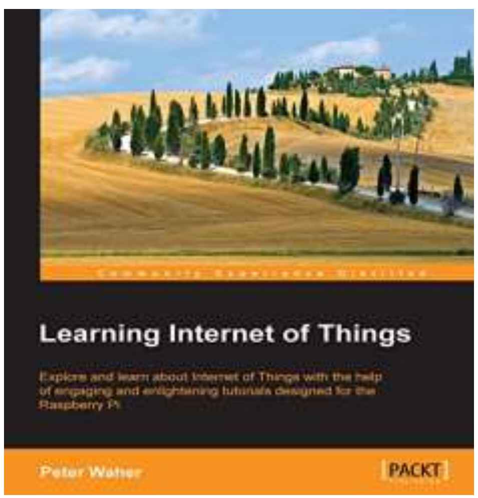
Resources


The Internet of Things
Applications to the Smart Grid and Building Automation
OLIVIER HERSENT | DAVID BOSWARTHICK | OMAR ELLOUMI
WILEY


Microsoft
Programming for the Internet of Things
Using Windows 10 IoT Core and Azure IoT Suite
Professional
Dawid Borycki

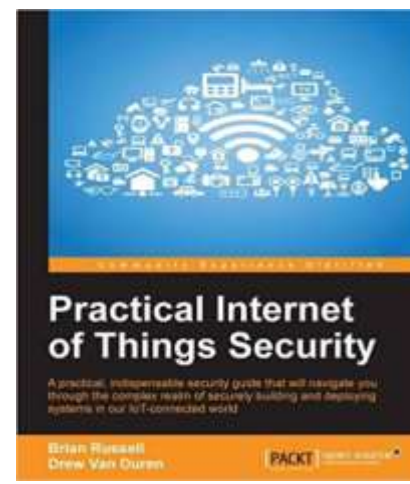

O'REILLY
Designing Connected Products
UX FOR THE CONSUMER INTERNET OF THINGS
Claire Rowland, Elizabeth Goodman, Martin Charlier, Ann Light & Alfred Lui
Foreword by Tom Igoe


IoT Solutions in Microsoft's Azure IoT Suite
Data Acquisition and Analysis in the Real World
Building for the Internet of Things
Scott Klein
Apress


Learning Internet of Things
Peter Waher
PACKT


Rubén Oliva Ramos
Internet of Things Programming with JavaScript
Learn the art of bringing the Internet of Things into your projects with the power of JavaScript
Packt


Internet of Things with Arduino Blueprints
PACKT


Practical Internet of Things Security
PACKT

IoT Use Cases In Agriculture

1. Monitoring of climate conditions
   - Weather stations
2. Greenhouse automation
3. Crop management
4. Cattle monitoring
5. Farm management systems

1

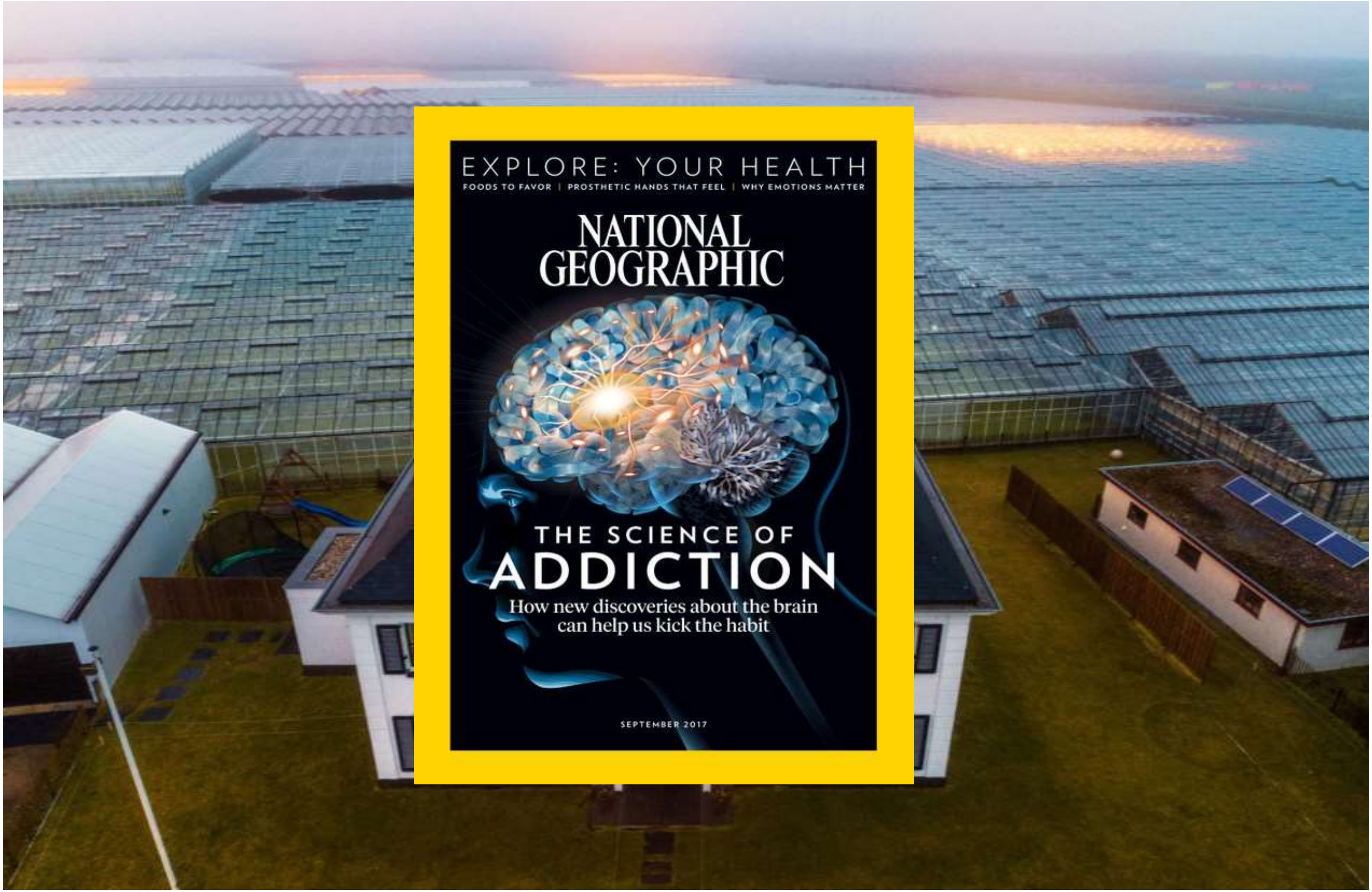
2

5

4

3

This Tiny Country (NL) Feeds the World



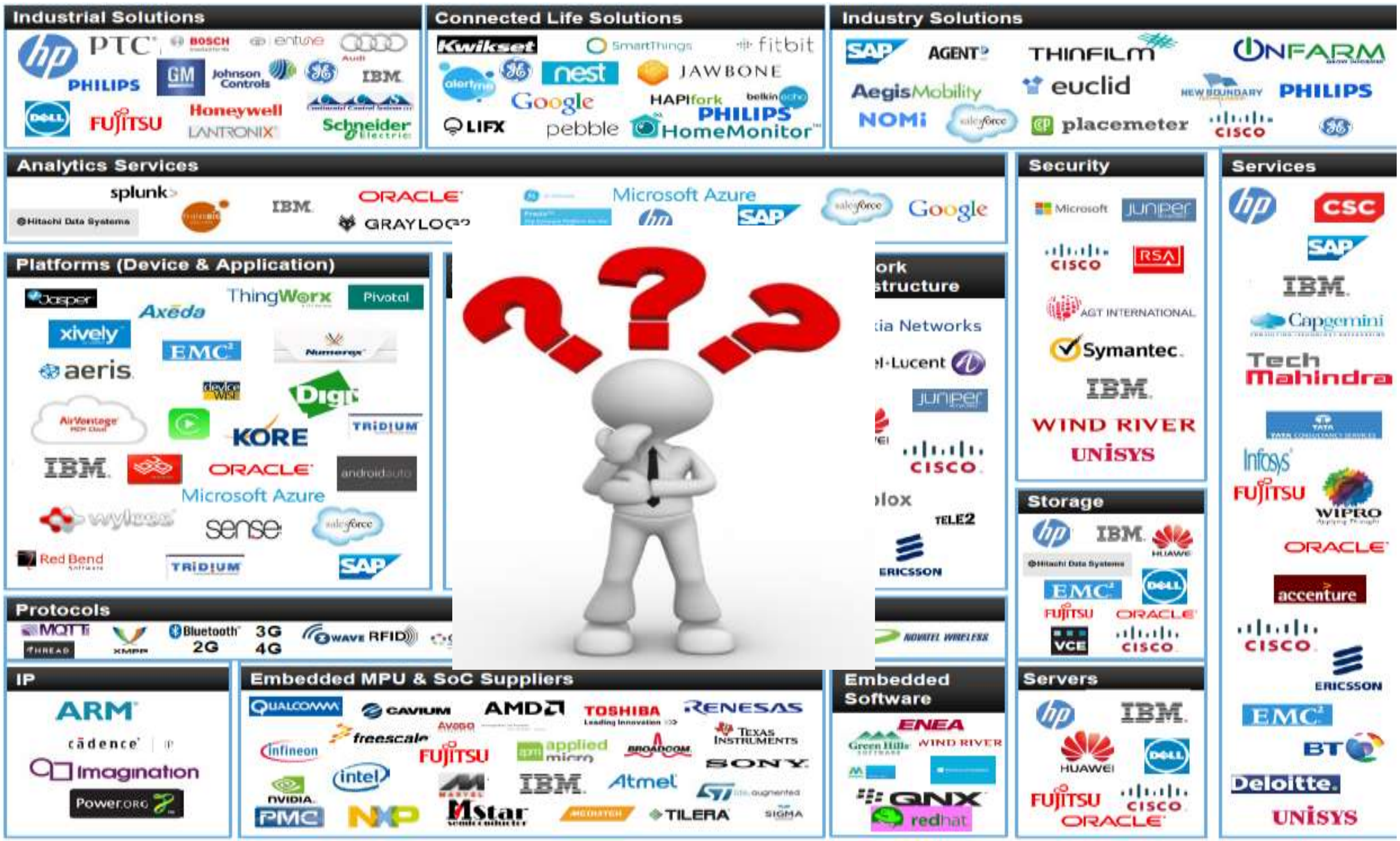
Crop growing around the clock...

A farm atop a former factory in the Hague

Innovative Approach

- **Sensors monitor**
  - CO_2 level
  - Humidity
  - LED light
  - Plant health
- **Data analytics**
  - Sustainable environment in the greenhouse
- Harvesting **10 times** the average yield

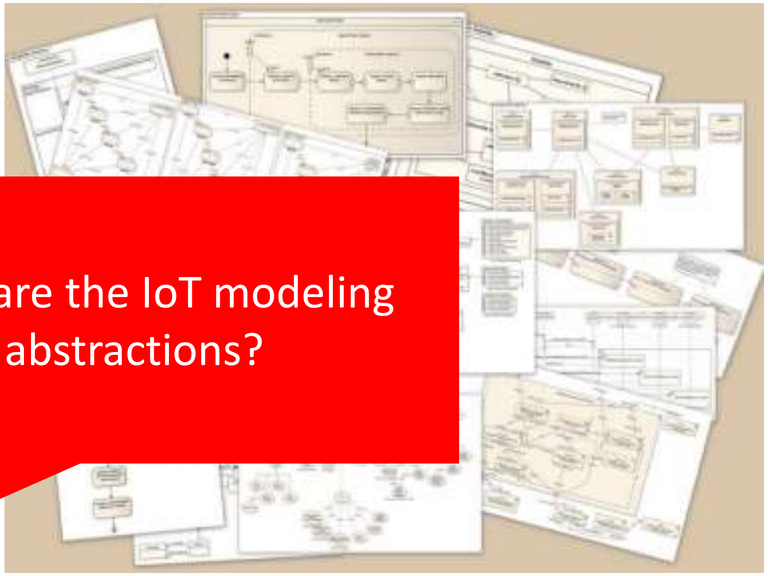
IoT Landscape



Software Engineering Models

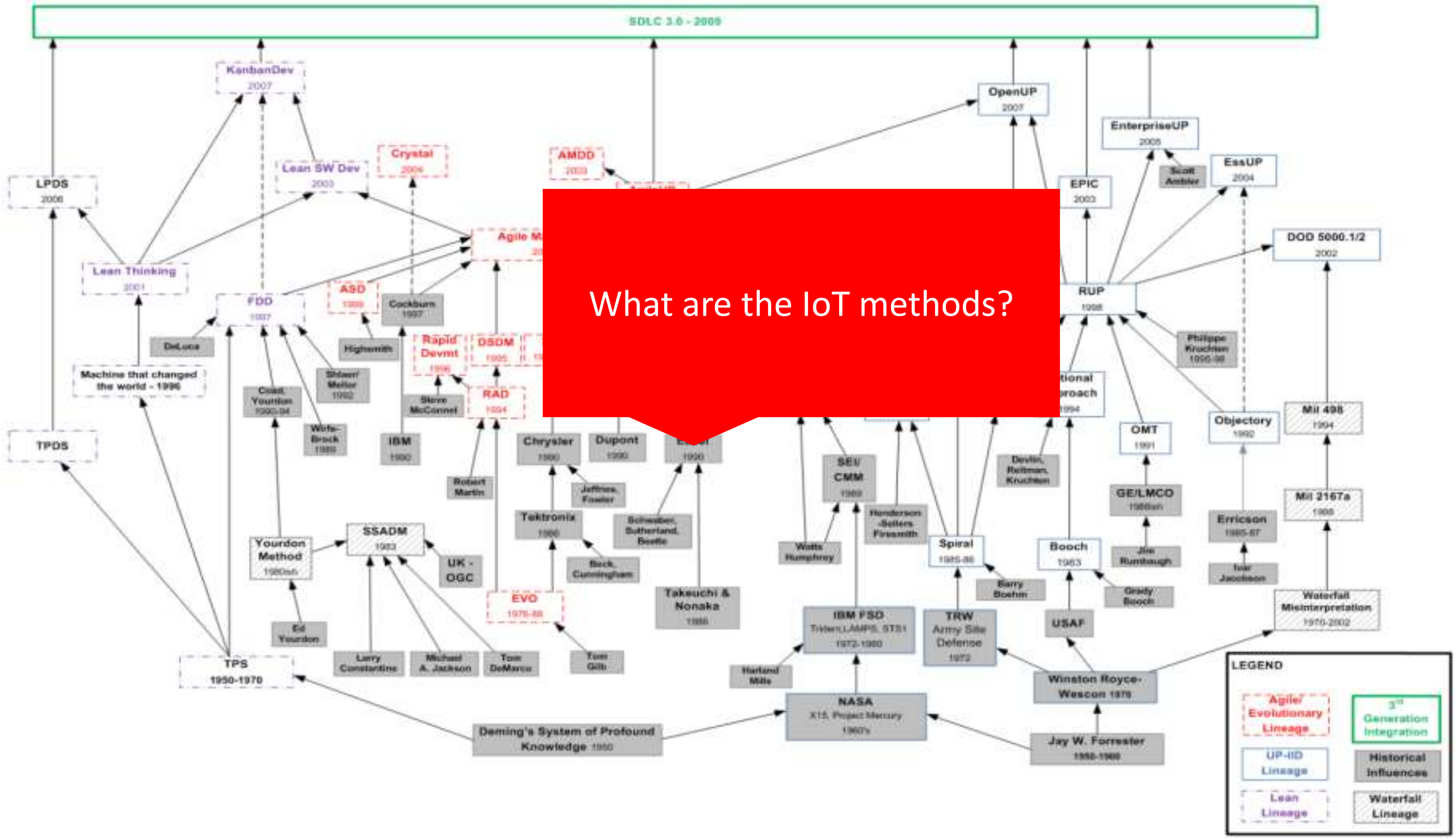
<u>The Unified Modeling Language</u>

Graphical ***language for visualizing, specifying, constructing, and documenting the artifacts*** *of a software-intensive system*



What are the IoT modeling abstractions?

Software Development Methods



- 32 methods emerged in the last 20 years

Ref: Mark Kennaley, SDLC 3.0: Beyond a Tacit Understanding of Agile, Fourth Medium Press, 2010.

IoT Methods

Method	Abbr.	Origin	Base
The Ignite | IoT Methodology (Slama et al. 2016)	Ignite	Industry	Best practices from the projects in the industry; Project Management Guidelines, such as PMBOK
The IoT Methodology (Collins)	IoT-Meth	Industry	Best practices from the projects in the industry
IoT Application Development (Patel and Cassou 2015), (Patel 2014)	IoT-AD	Academia	Built on macroprogramming approach; inspired by model-driven design
ELDAMeth (Fortino and Russo 2012), (Fortino et al. 2014), (Fortino et al. 2015), (ELDAMeth web site)	ELDAMeth	Academia	Multi-agent system development
A Software Product Line (SPL) Process to Develop Agents for the IoT (Ayala and Amor 2012), (Ayala et al. 2012), (Ayala et al. 2014), (Ayala et al. 2015)	SPLP-IoT	Academia	Multi-agent system development; Software Product Line Engineering
A General Software Engineering (SE) Methodology for IoT (Zambonelli 2016), (Zambonelli 2017)	GSEM-IoT	Academia	Traditional software development; Abstractions from IoT systems

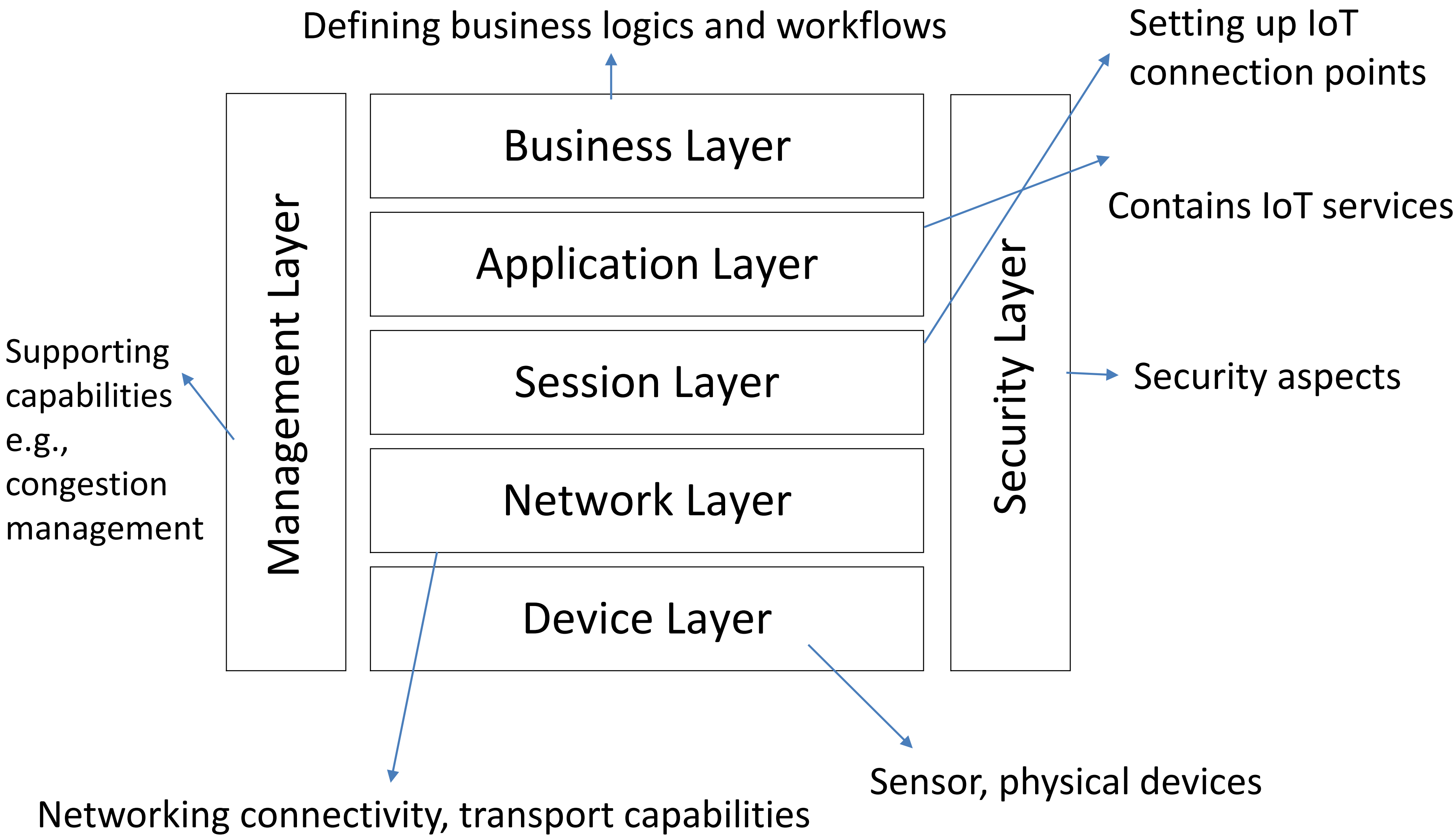
G. Giray, B. Tekinerdogan, E. Tüzün. ***Evaluation Framework for Characterizing IoT System Development Methods***, CRC Taylor and Francis, accepted for publication, in: Q. Hassan, A.R Khan, S. Madani (Eds.). Internet of Things Concepts, Technologies, Applications, and Implementations, CRC Press, 2017.

IoT Method Artefacts

Method	Artifacts	
Ignite	• Detailed idea sketch • Business case document • Project organization • Initial project plan • Problem statement • Stakeholder analysis • Site survey • Solution sketch • Project dimensions • Quantity structure	• Milestone plan • Process maps/Use cases • UI mockups • Domain model • Asset integration architecture • SOA landscape • Software architecture • Technical infrastructure • Hardware design
IoT-Meth	• IoT Canvas	• IoT-Architecture Reference Model
IoT-AD	• Vocabulary specification • Vocabulary framework • Architecture specification • Architecture framework • Application logic	• Deployment specification • Mapping files • Platform-specific device driver • Device-specific code
ELDAMeth	• Requirements • High-level design models • ELDA SO design models (Structural and behavioral) • Platform-independent ELDA SO code • Performance indices	• Simulator program • Platform-independent ELDA SO code • Simulation results • Platform-specific ELDA SO code • Test results
SPLP-IoT	• IoT multi-agent system variability model • IoT multi-agent system architecture • IoT multi-agent system application requirements	• Agent configuration • IoT multi-agent system architecture configuration • Refined goal models • Final application architecture
GSEM-IoT	no artifact defined	

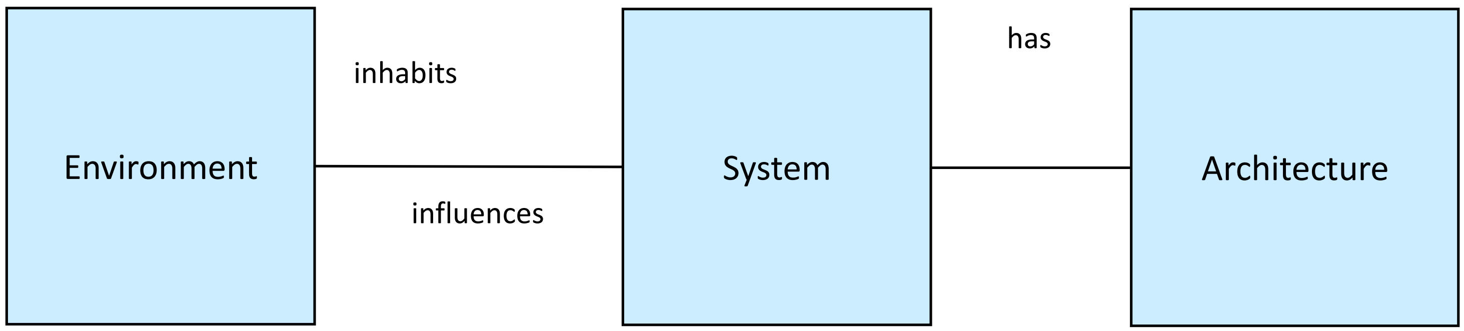
Architecture Modeling

IoT Reference Architecture



Software Architecture

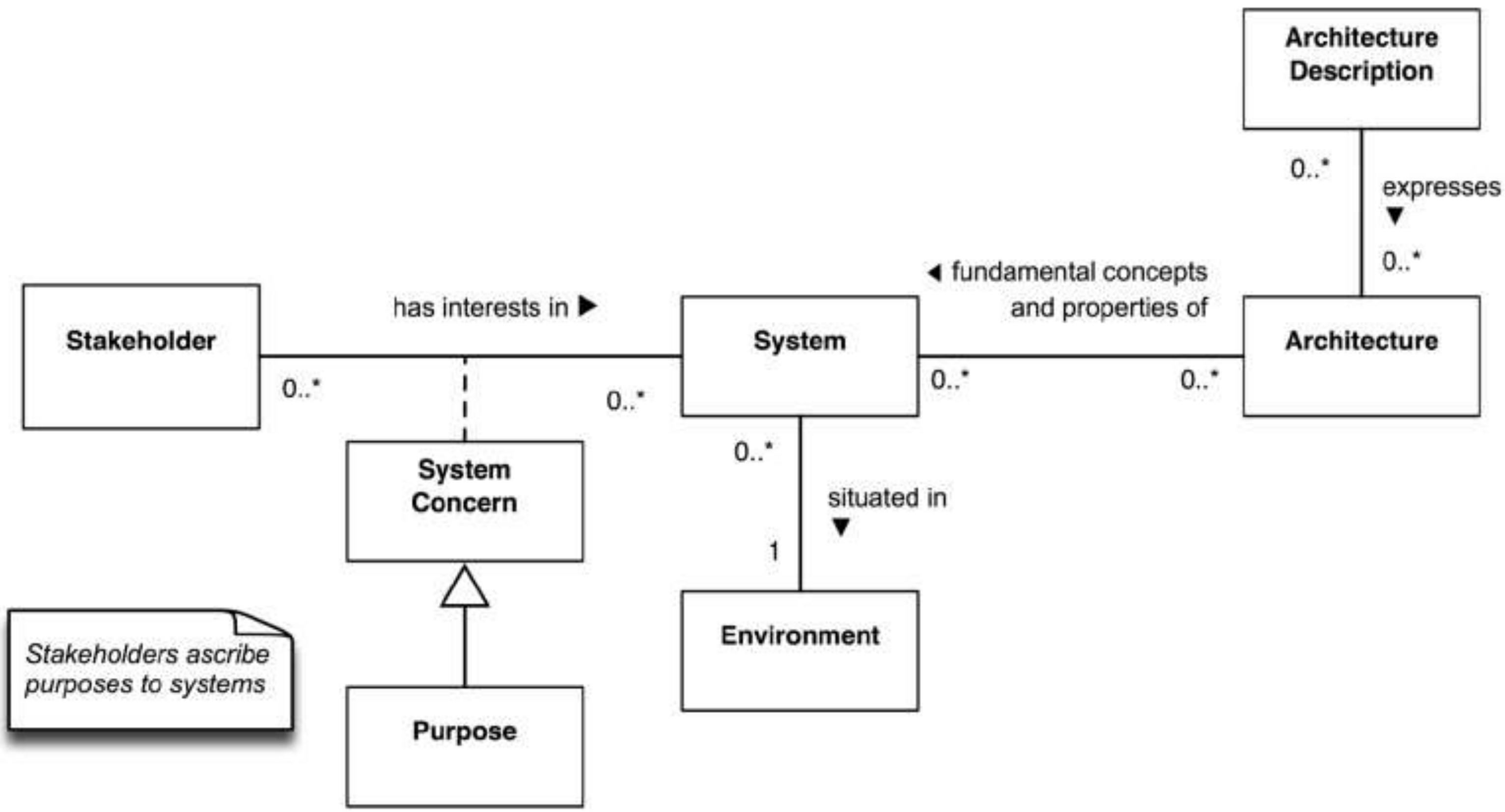
- *"Software architecture is the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution"*



IEEE. IEEE Product No. : SH94869-TBR: *Recommended Practice for Architectural Description of Software Intensive Systems*. IEEE Standard No. 1471-2000. Available at: http://shop.ieee.org/store/.

Architecture Stakeholders

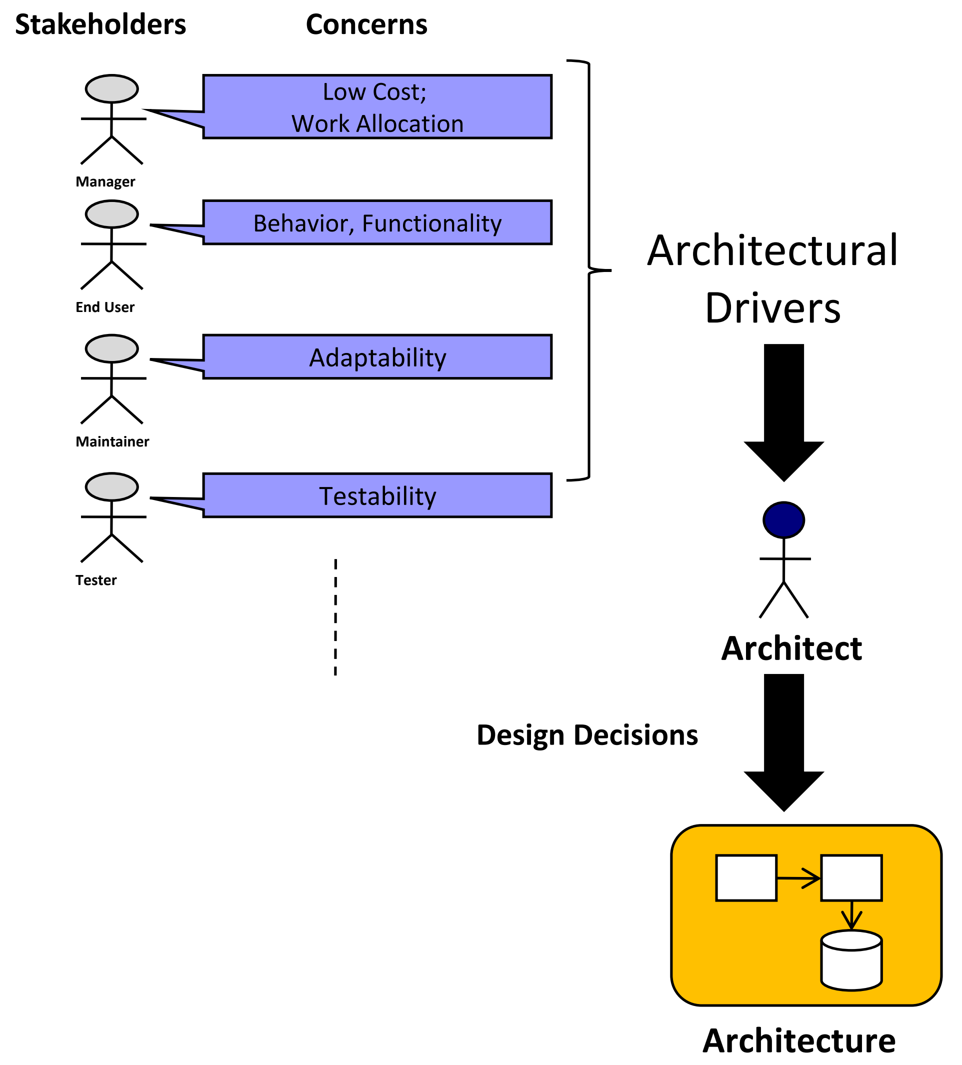
- **System Stakeholder:**
  an individual, team, or organization with interests in, or concerns relative to a system



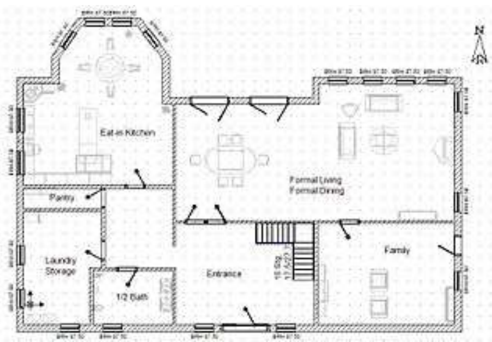
[ISO/IEC 42010:2007] Recommended practice for architectural description of software-intensive systems (ISO/IEC 42010) July 2007.

Architectural Drivers

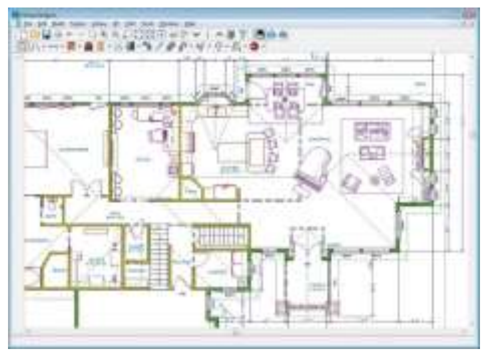
- **Stakeholder** is any person who has interest in the architecture
- Each stakeholder can have different **concerns**
- Each concern puts forces on the architect and influences the early design decisions that the architect makes



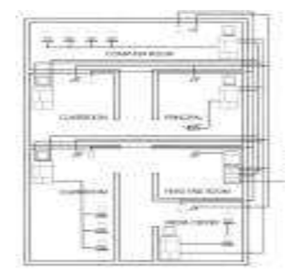
Multiple Views of the Architecture…



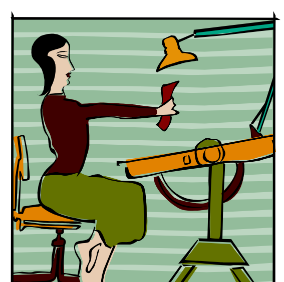
Floor plan



Interior Plan



Wiring Plan



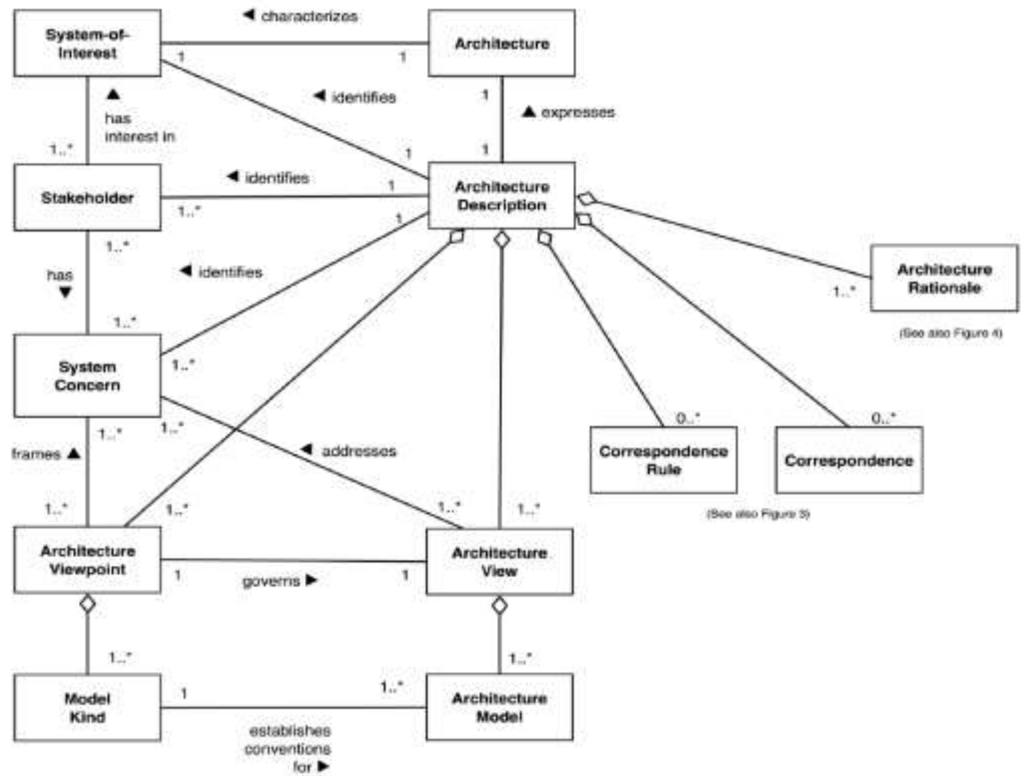
Architectural Viewpoints

View:

- a representation of a system from the perspective of one or more concerns which are held by one or more stakeholders

Viewpoint:

- A pattern or template from which to construct individual views

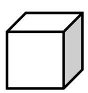


[ISO/IEC 42010:2007] Recommended practice for architectural description of software-intensive systems (ISO/IEC 42010) July 2007.

Example – UML Deployment Viewpoint

Viewpoint

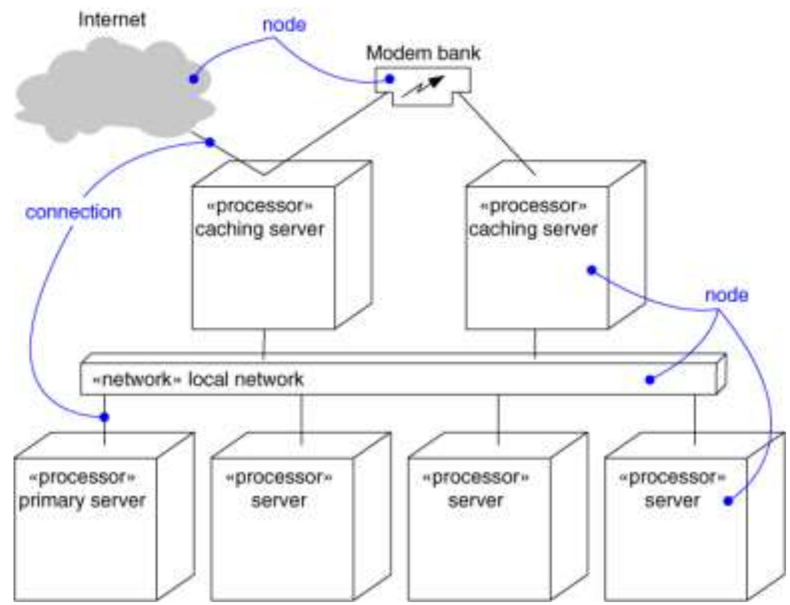
- Name: Deployment Viewpoint
- <u>Stakeholders:</u>
  - System Designer
- <u>Concerns:</u>
  - System Design
- <u>Components:</u>
  - Processing Nodes
- Notation



Node

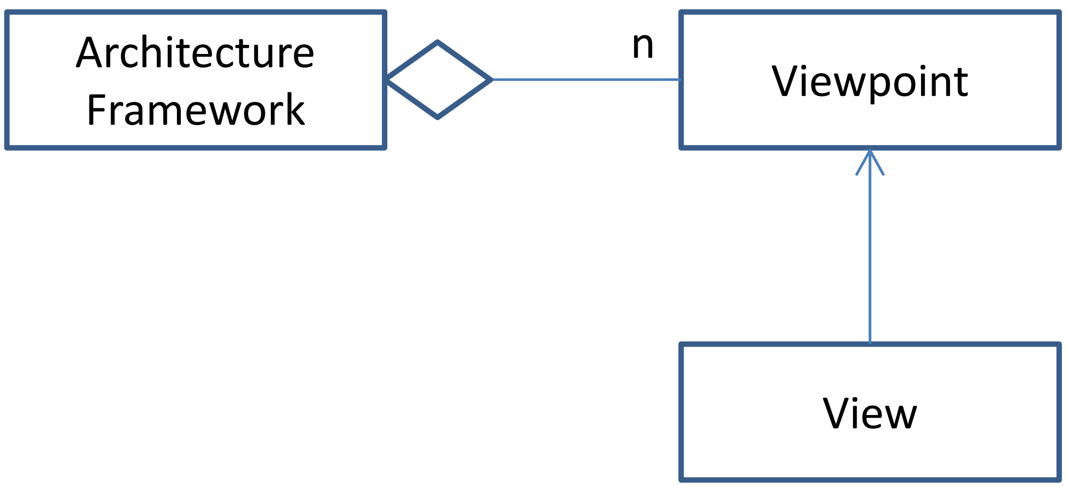
Connection

Deployment View - Example

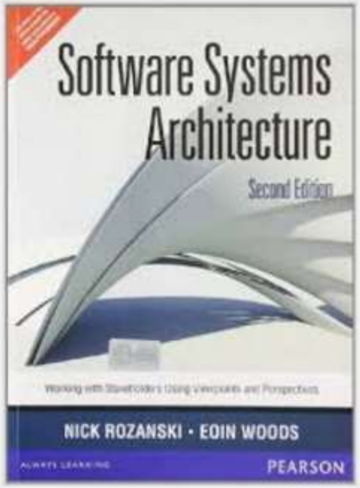


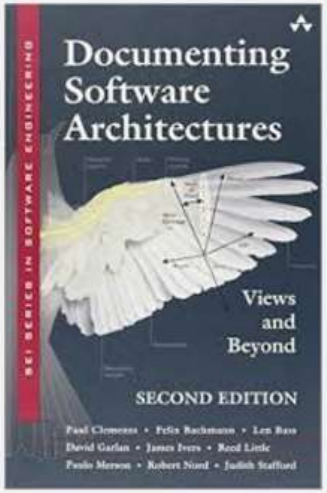
Architecture Framework

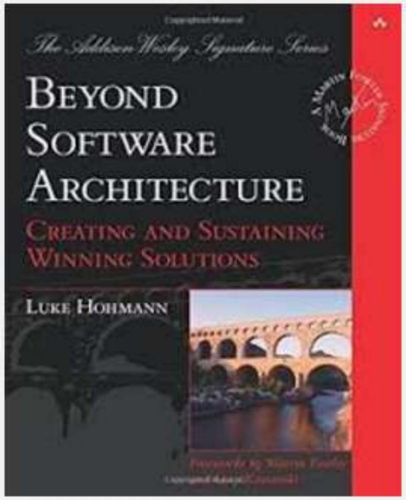
- ***Coherent set of viewpoints***
- Each viewpoint addressing single concern
- Separation of Concerns

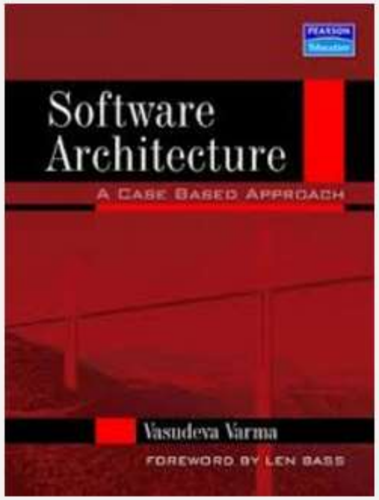


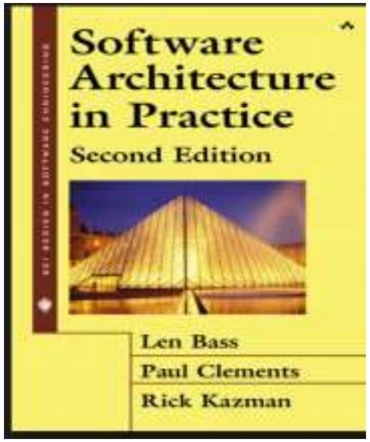
Architecture Viewpoint Approaches

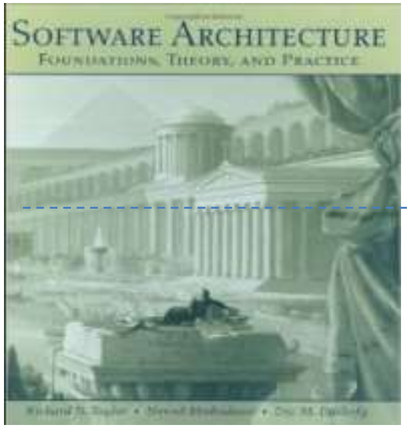












Architecture Frameworks

Last Updated: 09/11/2019 22:02:39

Survey of Architecture Frameworks

The specification of *architecture frameworks* is one area of standardization in ISO/IEC/IEEE 42010:2011 (the international revision of IEEE 1471:2000).

WG42 is collecting examples of architecture frameworks, listed below.

Corrections and additions may be sent to the webmaster. Please include: Name, Purpose, Scope, Brief Description and URL (or literature references).

Thanks to the following for their inputs: Takahiro Yamada (RASDS), Sly Gryphon (IFW), Alexander Ernst (EAM-PC), Nic Plum (TRAK), Graham Berrisford (Avancier), Kevin Smith (PEAF), Mark Paauwe (Dragon1), Danny Greefhorst (various), Christian Schweda (BPEAM), Nick Rozanski (RW), Andrew Guitarte (BCA), Neil C. Greenfield (SABSA), Daniele Gianni (ESA), Jose L. Fernandez (PPOOA), Patrizio Pelliccione (MEGAF), Roger Evernden (IFW), Vanessa Douglas-Savage (QGEA), Adrian Grigoriu (FFLV+GODS), Dan Haner (DODAF), David Sprott (CBDI-SAE), Anders W. Tell, Harris Veziris (eTOM), Ger Schoeber (CAFCR), Bruce McNaughton (AF4Orgs, AF4MgtSys), Damian Andrew Tamburri (SQUID), Joris van den Aker (CAFCR+), Ian Glossop (ARIES), Phil Cutforth (GEA-NZ), Mathias Axling (RAMI4.0).

ID	Name	Purpose	Scope	Classifiers	Notes
AF-EAF	Air Force Enterprise Architecture Framework	"The AF Enterprise Architecture Framework (AF-EAF) provides a logical structure for classifying, organizing and relating the breadth and depth of information that describes and documents the Air Force Enterprise Architecture (AF-EA)."	Air Force IT systems	Communication, Guidance, Enterprise Architecture Descriptions	"The AF-EAF does not define the AF-EA content, rather it consists of various approaches, models, and definitions for communicating and facilitating the presentation of key architecture components (i.e. architecture vision, governance, principles, guidance, products, etc.) required for the development and integration of AF architectures. The AF-EAF establishes a common foundation for understanding, comparing and integrating architectures and as such provides the overarching guidance for generating AF architectures." [All quotes from AF-EAF v2.01, 6 June 2003]
AFIoT	IEEE P2413 – Architecture Framework for the Internet of Things	"This standard defines an architectural framework for the Internet of Things (IoT), including descriptions of various IoT domains, definitions of IoT domain			"The architectural framework for IoT provides a reference model that defines relationships among various IoT verticals (e.g., transportation, healthcare, etc.) and common architecture elements. It also provides a blueprint for data abstraction and the quality 'quadruple' trust that includes protection, security, privacy, and

>70 AFs!

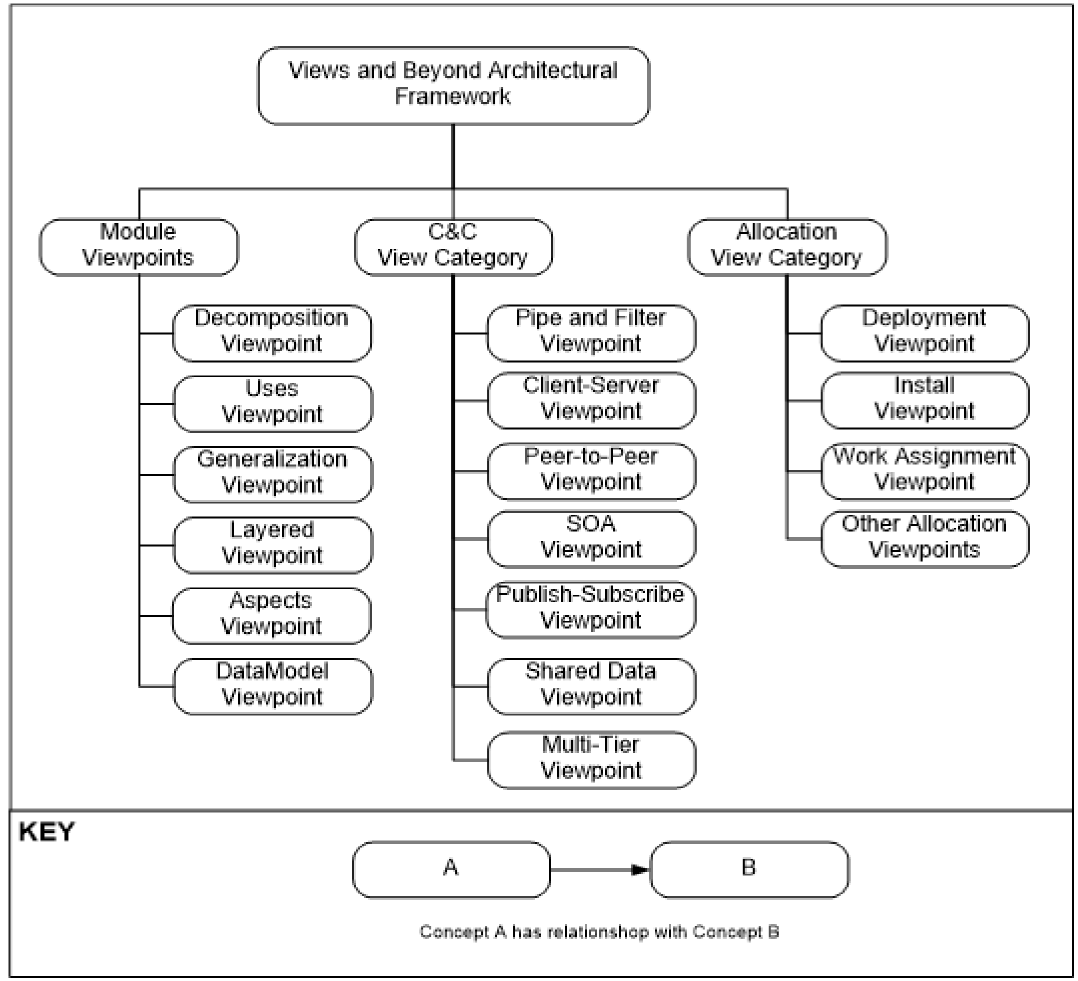
http://www.iso-architecture.org/42010/afs/frameworks-table.html

Example Architecture Framework

- **Module Styles**
  - How is the architecture structured as a set of **implementation units**?

- **Component-and-Connector Styles**
  - How is the architecture structured as a set of **execution units**?

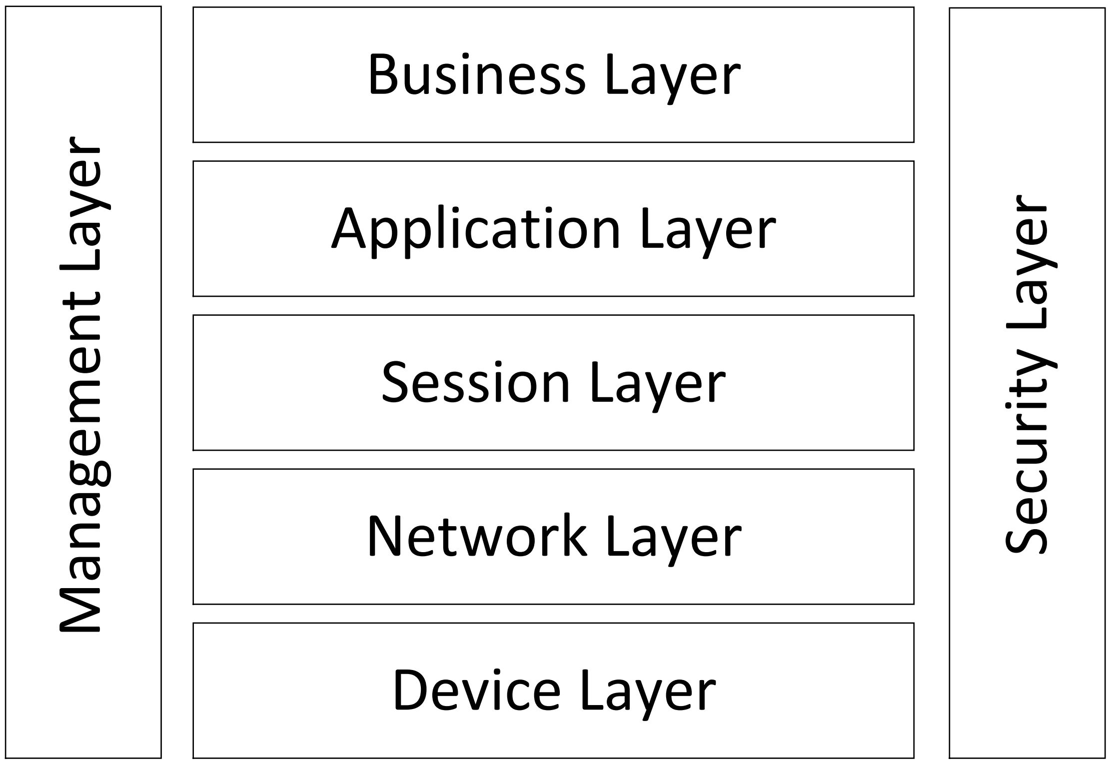
- **Allocation Styles**
  - How does the architecture **relate to non-software structures** in its environment?



P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord, J. Stafford. Documenting Software Architectures: **Views and Beyond**. Second Edition. Addison-Wesley, 2010

Architecture Design of IoT

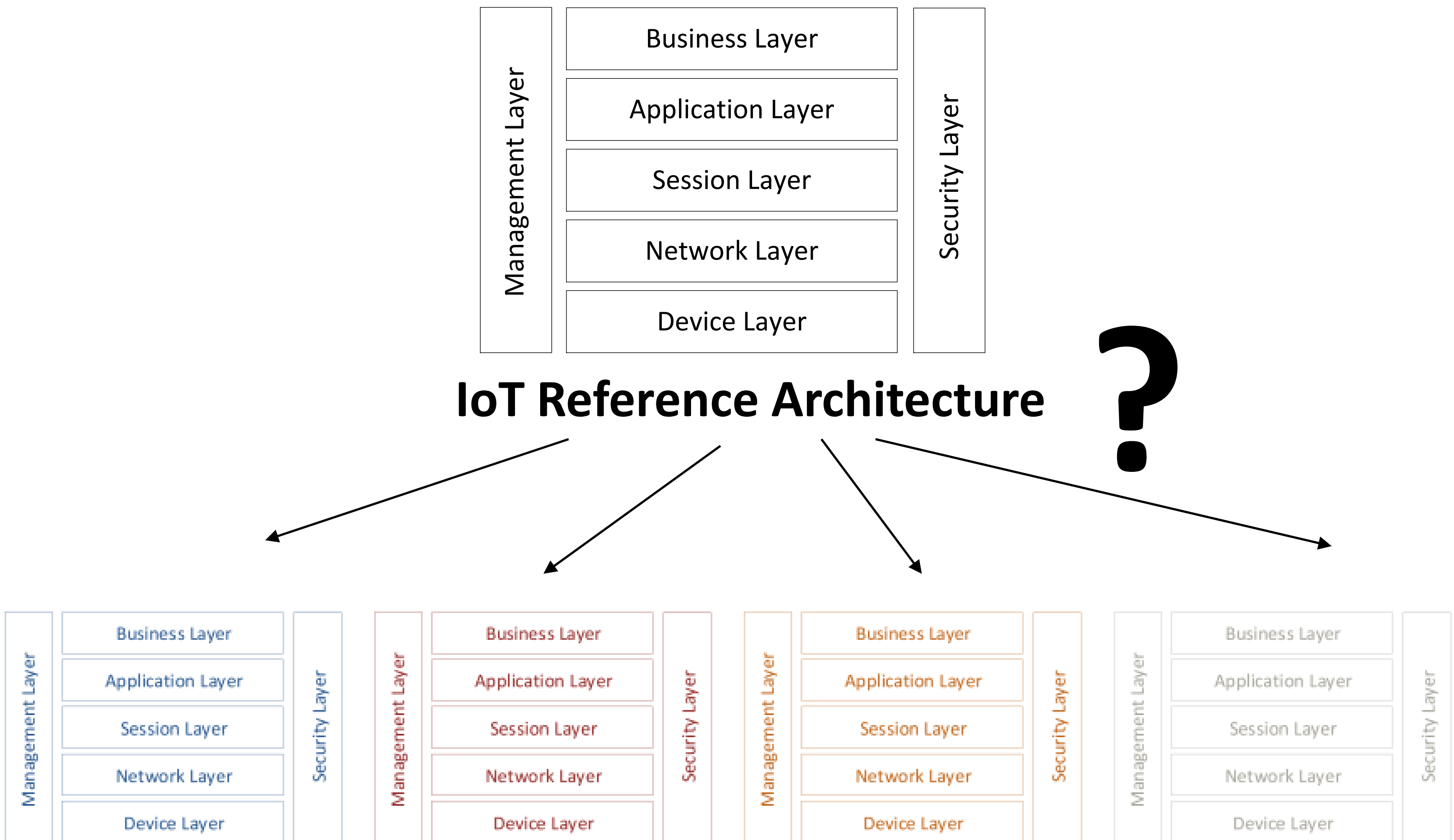
IoT Reference Architecture



Application Domains…

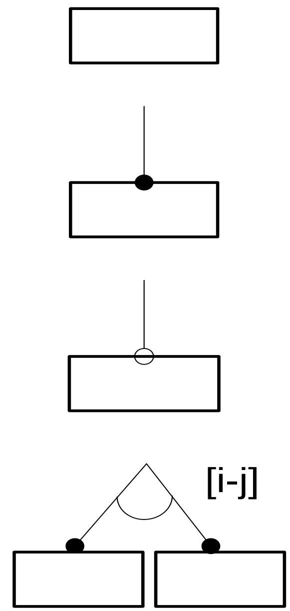
- **Consumers**
  - Home automation, wearable, Health and Wellness,...)
- **Commercial**
  - Logistics, Retail, Building
- **Industrial**
  - Manufacturing, Energy, Transportation,....
- **Public Sector**
  - Smart Cities and regions, Public Safety, Security, Healthcare

Reference Architecture vs. Application Architecture



Feature-Oriented Domain Modeling

- **Feature model**
  - common and the variable features of products and the dependencies between the variable features
- **Feature**
  - a distinctive property of a concept (domain model)
  - user visible characteristic of a system (requirements)
- **Feature diagram**
  - consists of a set of nodes, a set of directed edges, and a set of edge decorations

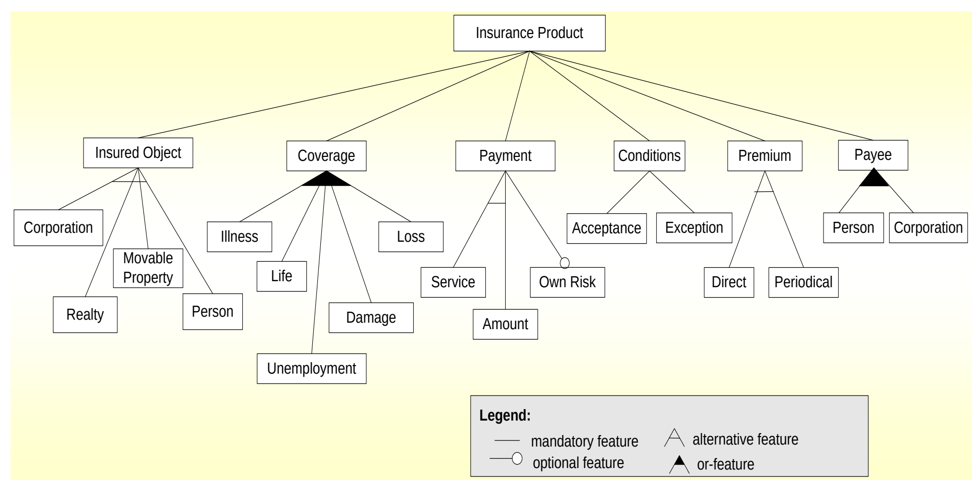


Feature

Mandatory feature

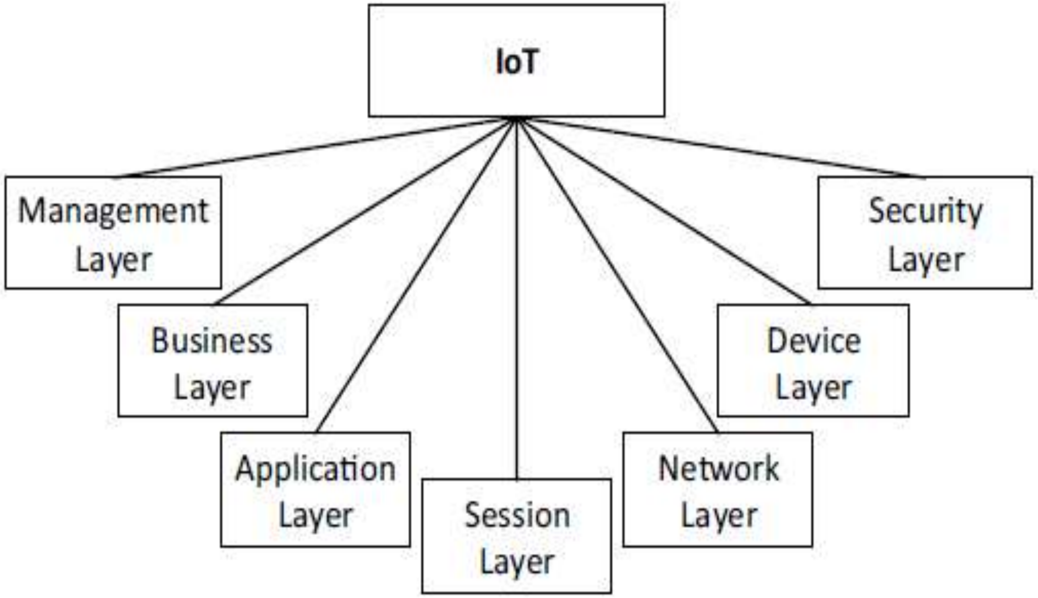
Optional feature

Feature group with cardinality i– j



Feature Model of IoT Protocols

- Top Level Feature Diagram



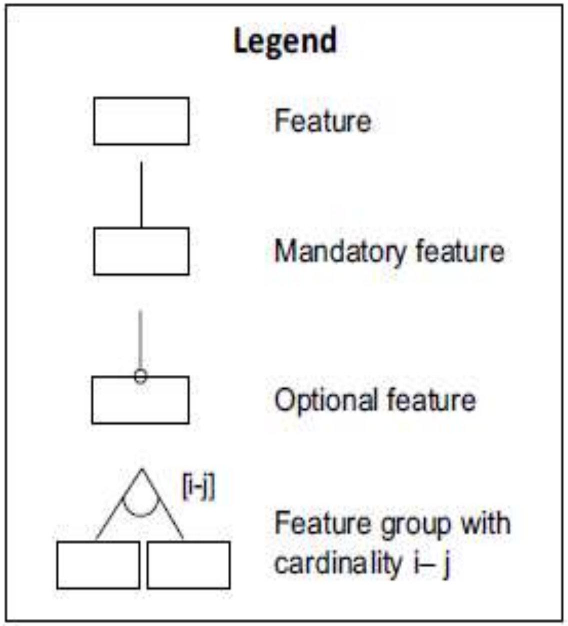
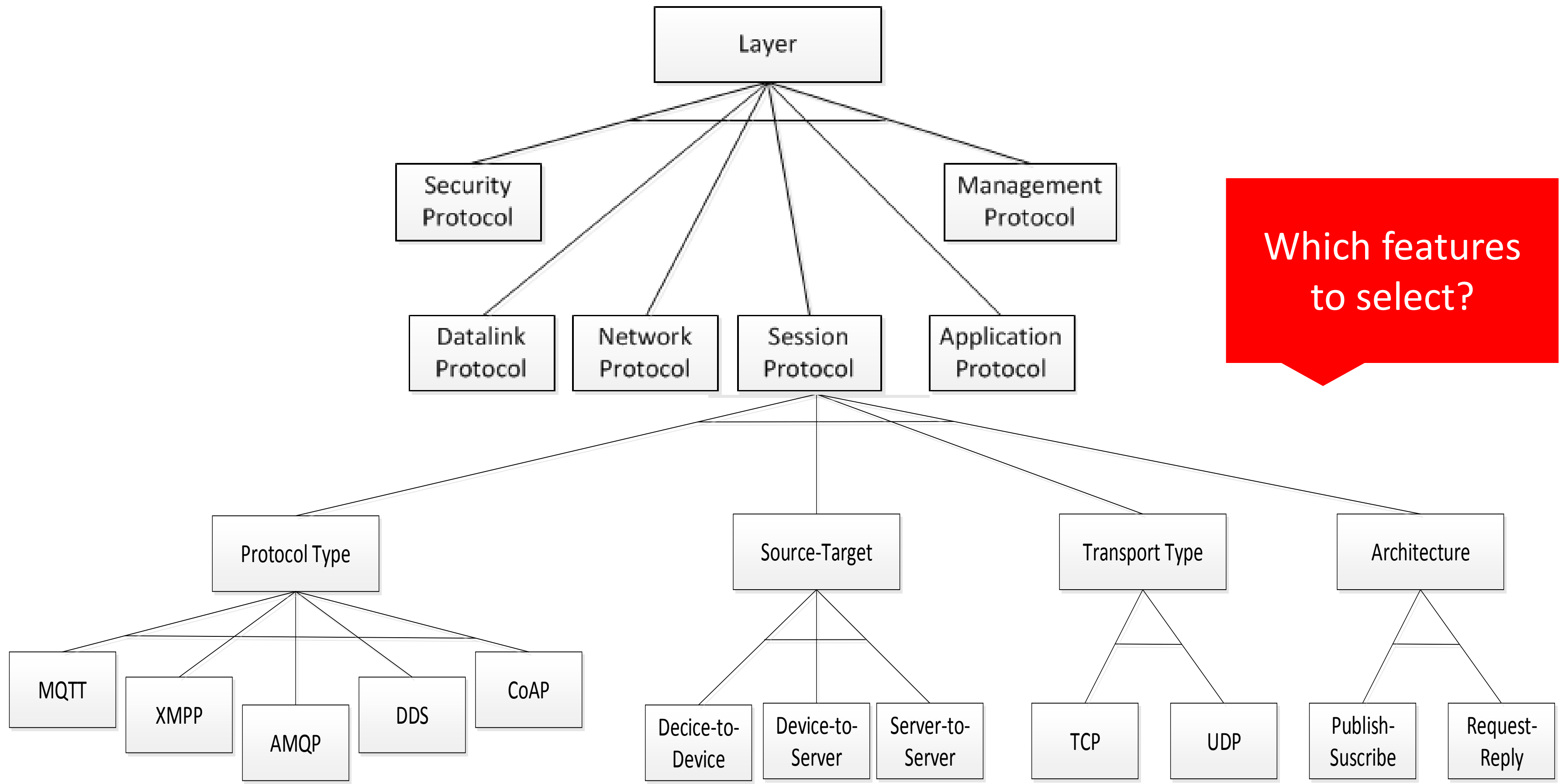


Fig. 3 Top level feature diagram of IoT

Feature Diagram of Session Layer Protocols



Which features to select?

Machine-to-Machine (M2M), Machine-to-Cloud (M2C), and Cloud-to-Cloud (C2C)

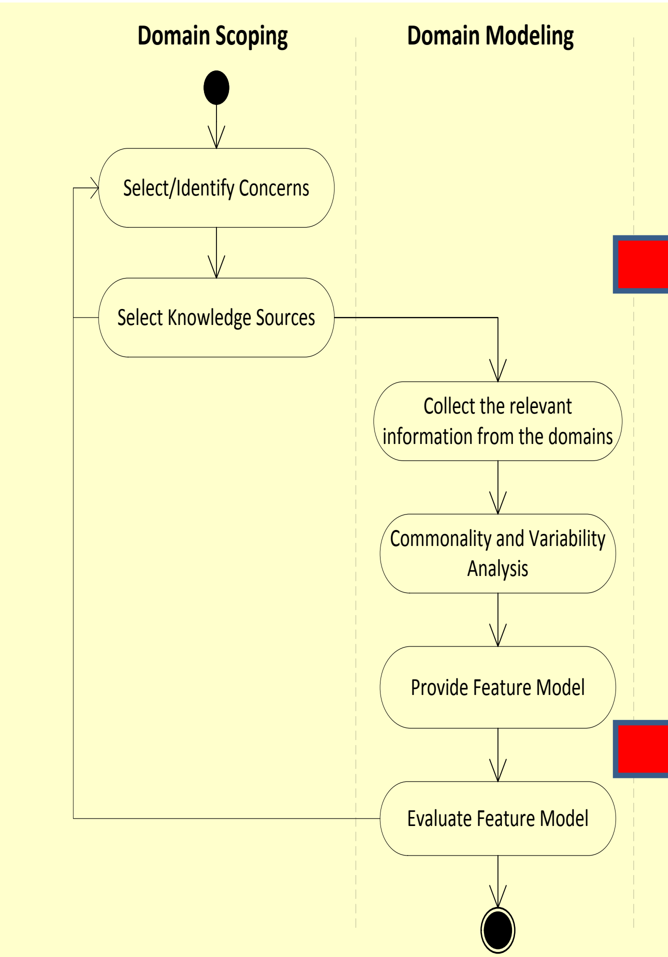
Evaluation Framework

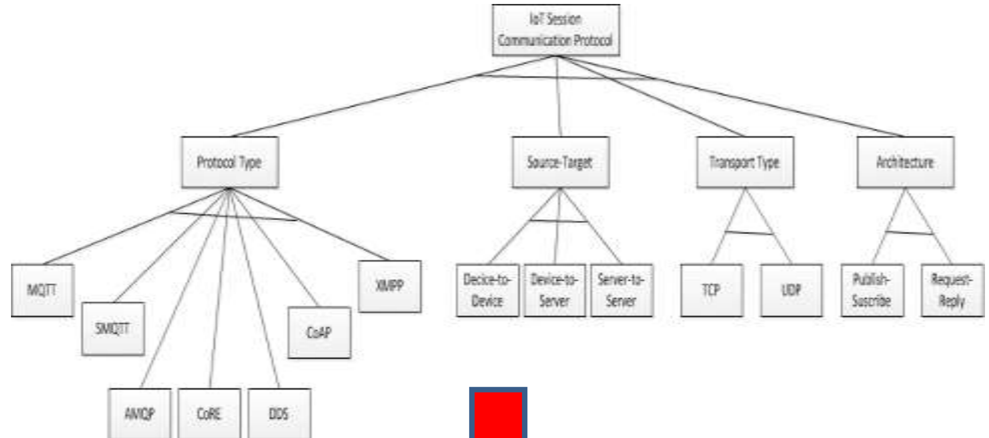
Criteria	Description	Possible Answers
Standard Organization	What is the standardization organization?	IETF, OASIS, OMG, …
Source - Target	What are the possible source-target relations?	D2D, D2S, S2S
Real-Time	Does the architecture allow real-time communication?	Yes/No/Partially
Brokered or Bus-Based Architecture	Is the architecture brokered or bus-based (unbrokered)?	Brokered/Unbrokered/Bus-Based
Communication Pattern	The adopted communication pattern including Pub/Sub or Request/Reply.	Pub/Sub, Request-Reply
Interoperable	What is the level of interoperability?	Syntactic, Semantics, …
Transport	What is the transport protocol (TCP/UDP)?	TCP, UDP, TCP+UDP
QoS	Are the quality of service parameters defined?	Yes, No, Partially
License	What is the license level of the communication protocol?	Open Source. Commercial
Mobile Support	Does the system provide mobile support?	Yes, No
Security	What is the adopted security protocol?	TLS/SSL, AES/HMAC-SHA, DTLS, …
Message/Data Centric	Is the protocol message or data centric?	Message Centric, Data Centric
Web/App. Based	Is the protocol web-based or application-based?	Web-based, Application-based

Adopted Criteria for Selecting Communication Protocol

CHARACTERISTICS	MQTT	XMPP	AMQP	DDS	CoAP
Standard	OASIS	IETF	OASIS	OMG	IETF
Source - Target	D2S	D2S	S2S	D2D	D2D
Real-Time	No	Near RT	No	Yes	No
Broker/Bus Based	Broker based	Bus Based	Broker based	Bus Based	Broker based
Com. Pattern	Pub/Sub	Pub/Sub	Pub/Sub	Pub/Sub	Request-Reply
Interoperable	Partial	Yes	Yes	Yes	Yes
Transport	TCP	TCP	TCP	TCP/UDP	UDP
QoS	Yes	Yes	Yes	Yes	Yes
License	Open Source & Commercial	Open Source & Commercial	Open Source & Commercial	Open Source & Commercial	Open Source & Commercial
Mobile Support	Yes	Yes	Yes	Yes	Yes
Security	TLS/SSL	TLS/SSL	TLS/SSL	AES/HMAC-SHA	DTLS
Message/Data Centric	Message	Data	Message	Data	Data
Web/App. Based	Application Based	Application Based	Application Based	Application Based	Web Based

Evaluation Framework





Criteria	Description	Possible Answers
Standard Organization	What is the standardization organization?	IETF, OASIS, OMG, ...
Source - Target	What are the possible source-target relations?	D2D, D2S, S2S
Real-Time	Does the architecture allow real-time communication?	Yes/No/Partially
Brokered or Bus-Based Architecture	Is the architecture brokered or bus-based (unbrokered)?	Brokered/Unbrokered
Communication Pattern	The adopted communication pattern including Pub/Sub or Request/Reply.	Pub/Sub, Request-Reply
Interoperable	What is the level of interoperability?	Syntactic, Semantics, ...
Transport	What is the transport prototocl (TCP/UDP)?	TCP, UDP, TCP+UDP
QoS	Are the quality of service parameters defined?	Yes, No, Partial
License	What is the license level of the communication protocol?	Open Source. Commercial
Mobile Support	Does the system provide mobile support?	Yes, No
Security	What is the adopted security protocol?	TLS/SSL, AES/HMAC-SHA, DTLS, ...
Message/Data Centric	Is the protocol message or data centric?	Message Centric, Data Centric
Web/App. Based	Is the protocol web-based or application-based?	Web-based, Application-based

O. Koksal, B. Tekinerdogan, Feature-Driven Domain Analysis of Session Layer Protocols of Internet of Things, in: Proc. of the 2nd. IEEE International Congress of Internet of Things, Honolulu, Hawaii, USA, June 25-June 30, 2017.

How to use the Evaluation Framework?

- The predefined criteria
  - select the proper communication protocol based on the requirements
- Based on the analysis
  - practitioners might select the feasible protocol
- In case more than one protocol is feasible
  - additional functional and non-functional requirements considered
- Number of protocols as well as evaluation criteria
  - can be extended

Feature Model for
Farm Management Information Systems (FMIS)

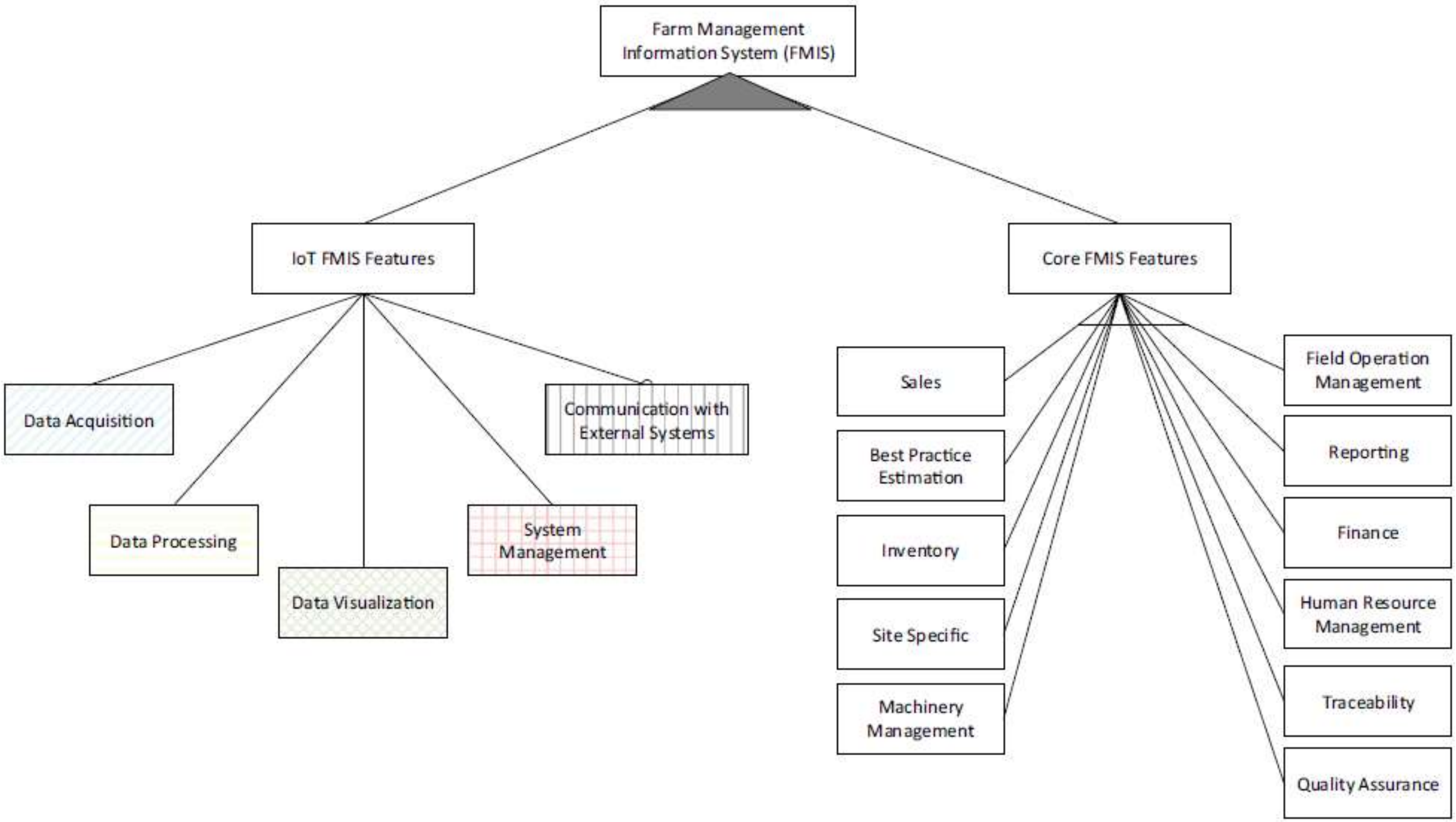


Fig. 5 Top level feature diagram of FMIS

IoT-based FMIS

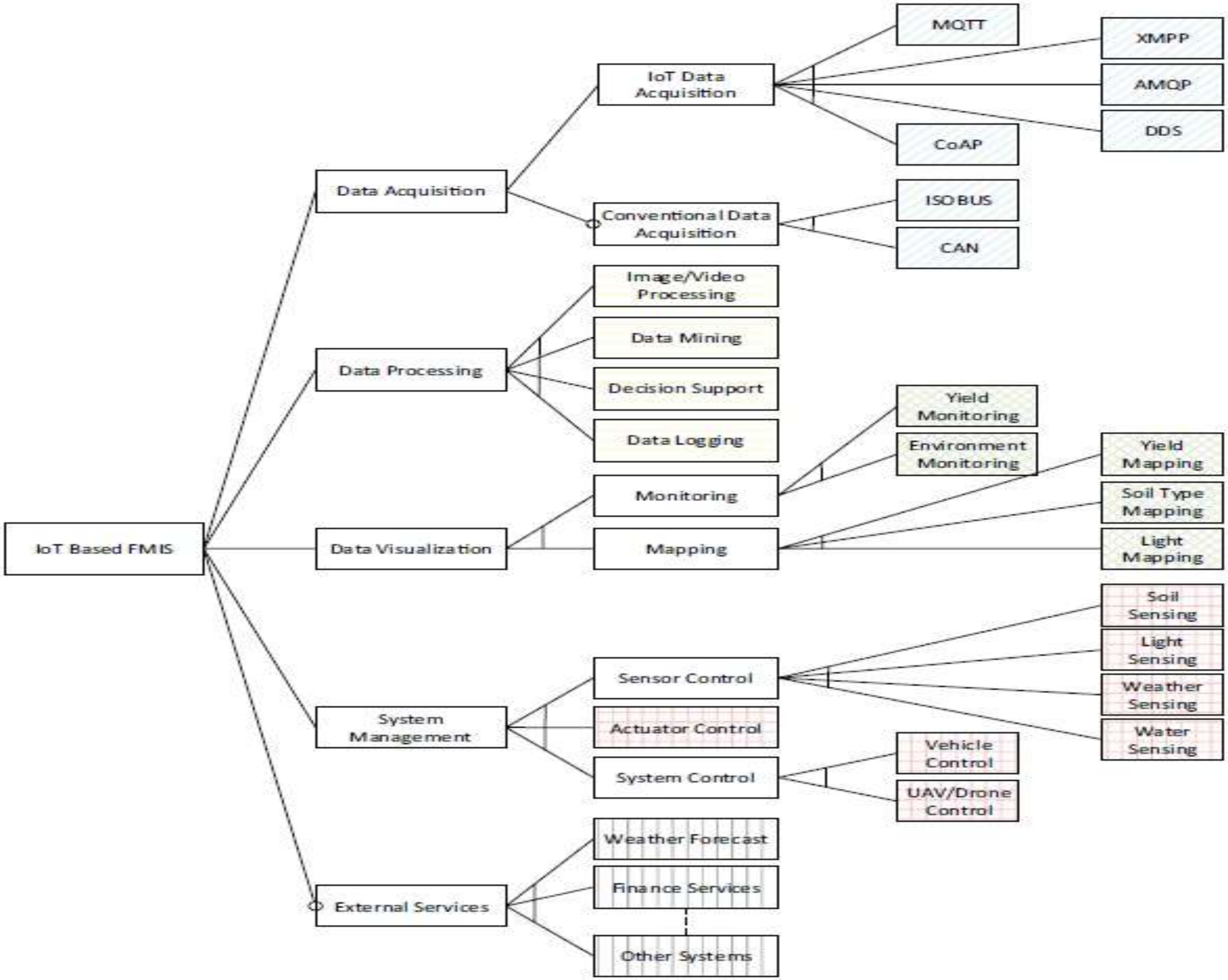


Fig. 6 Family feature diagram of FMIS software

Reference architecture for IoT-based FMIS
1. Decomposition View

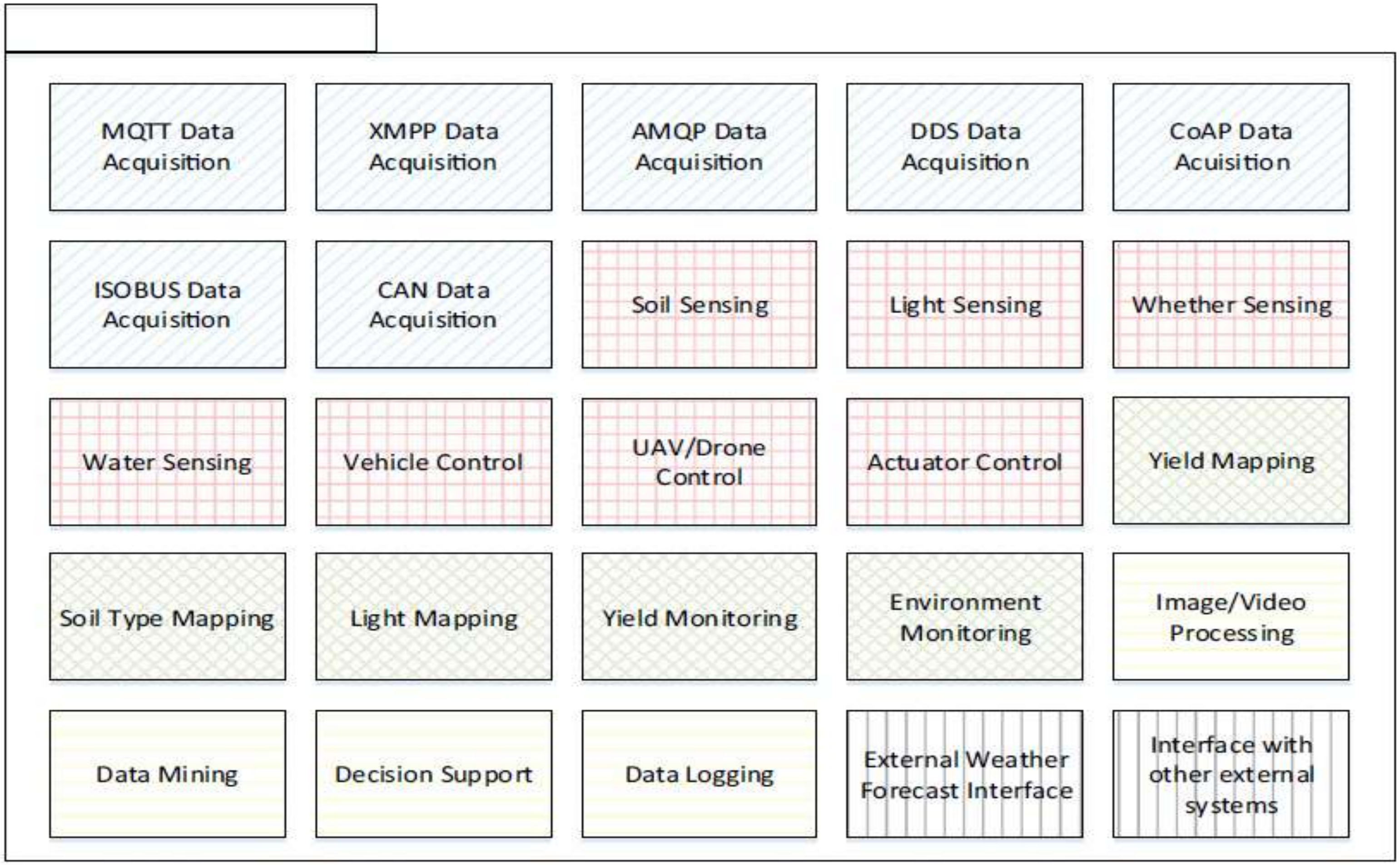


Fig. 7 IoT based FMIS—decomposition view

2. Layered View

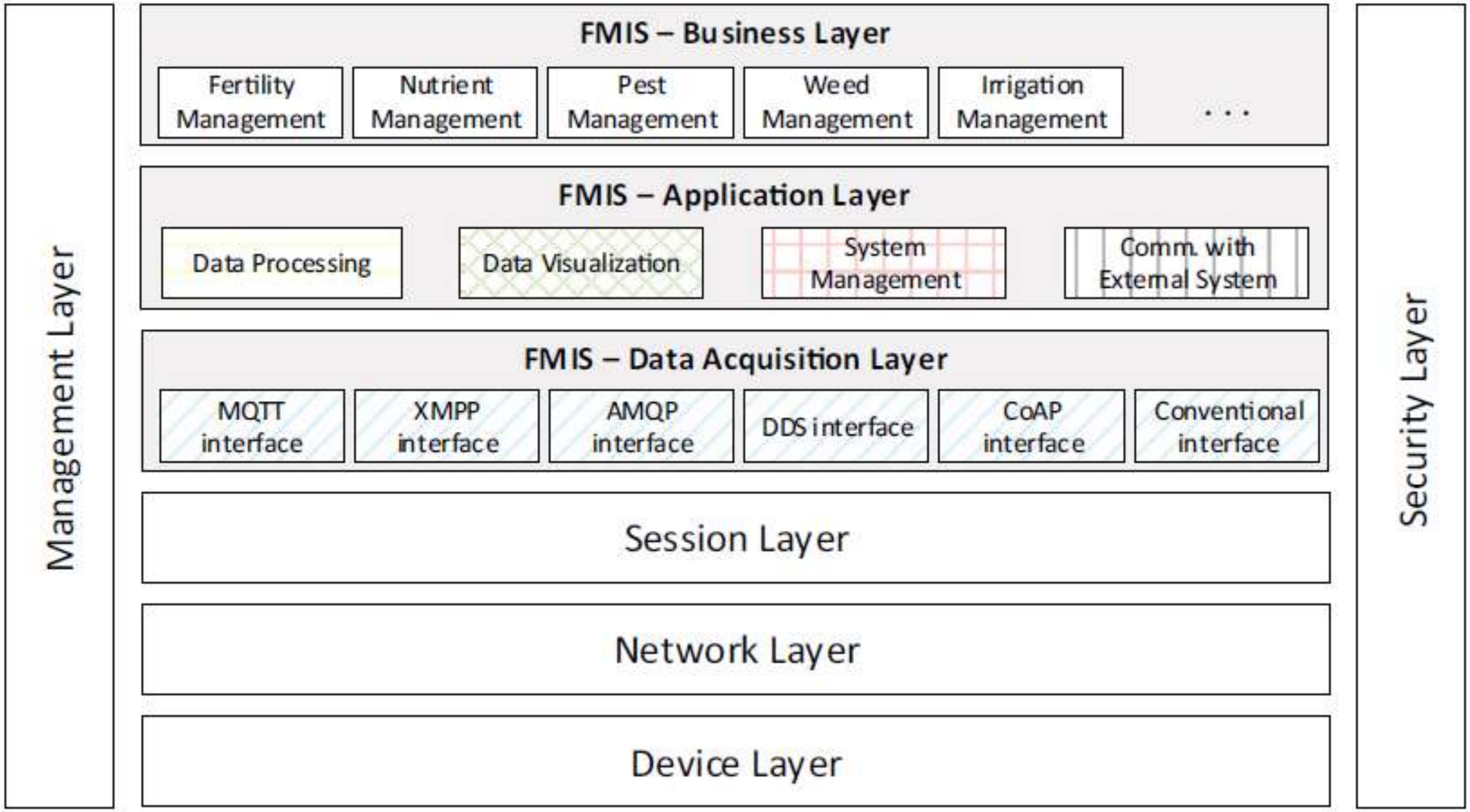


Fig. 8 IoT based FMIS—layered view

3. Deployment View

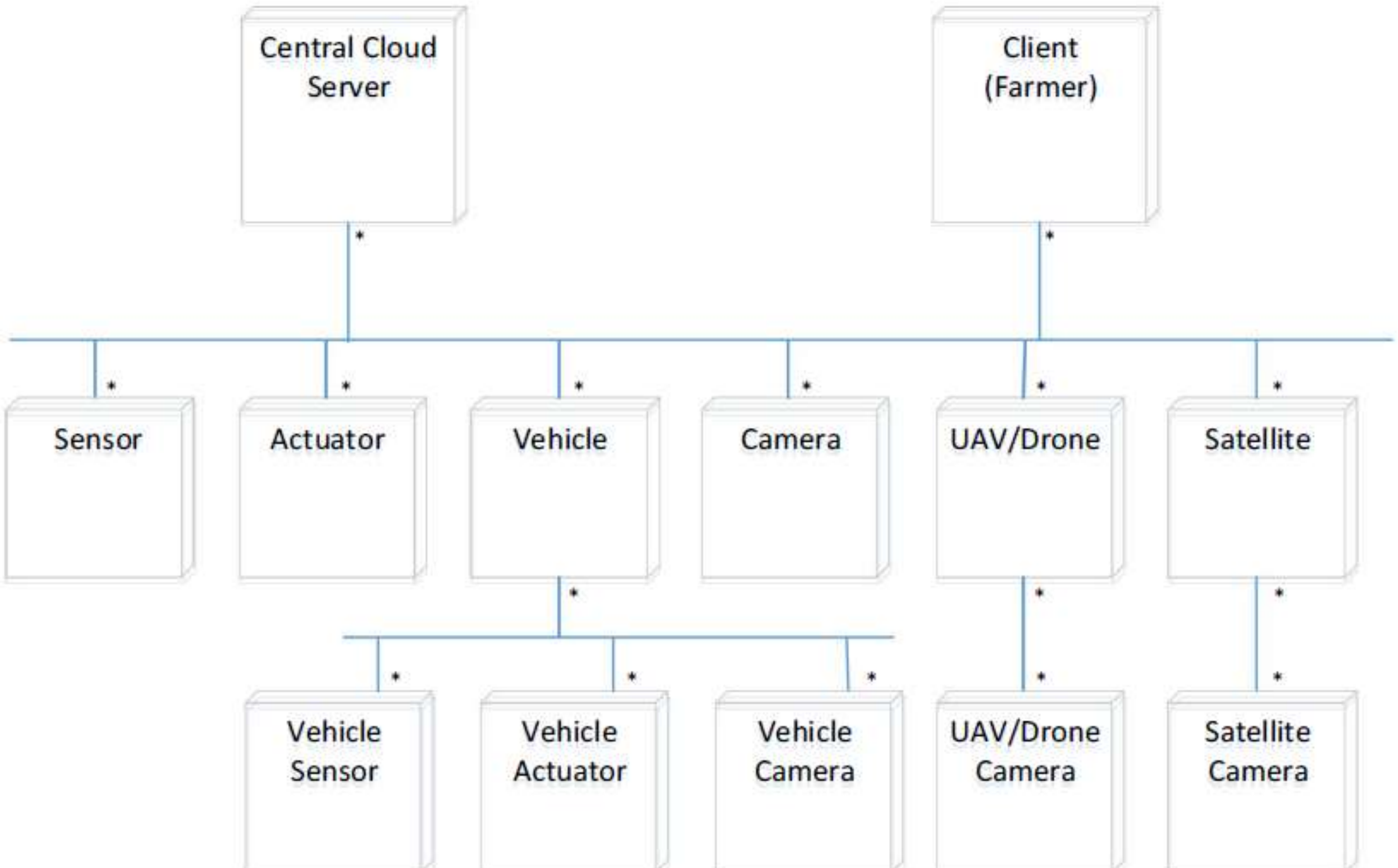
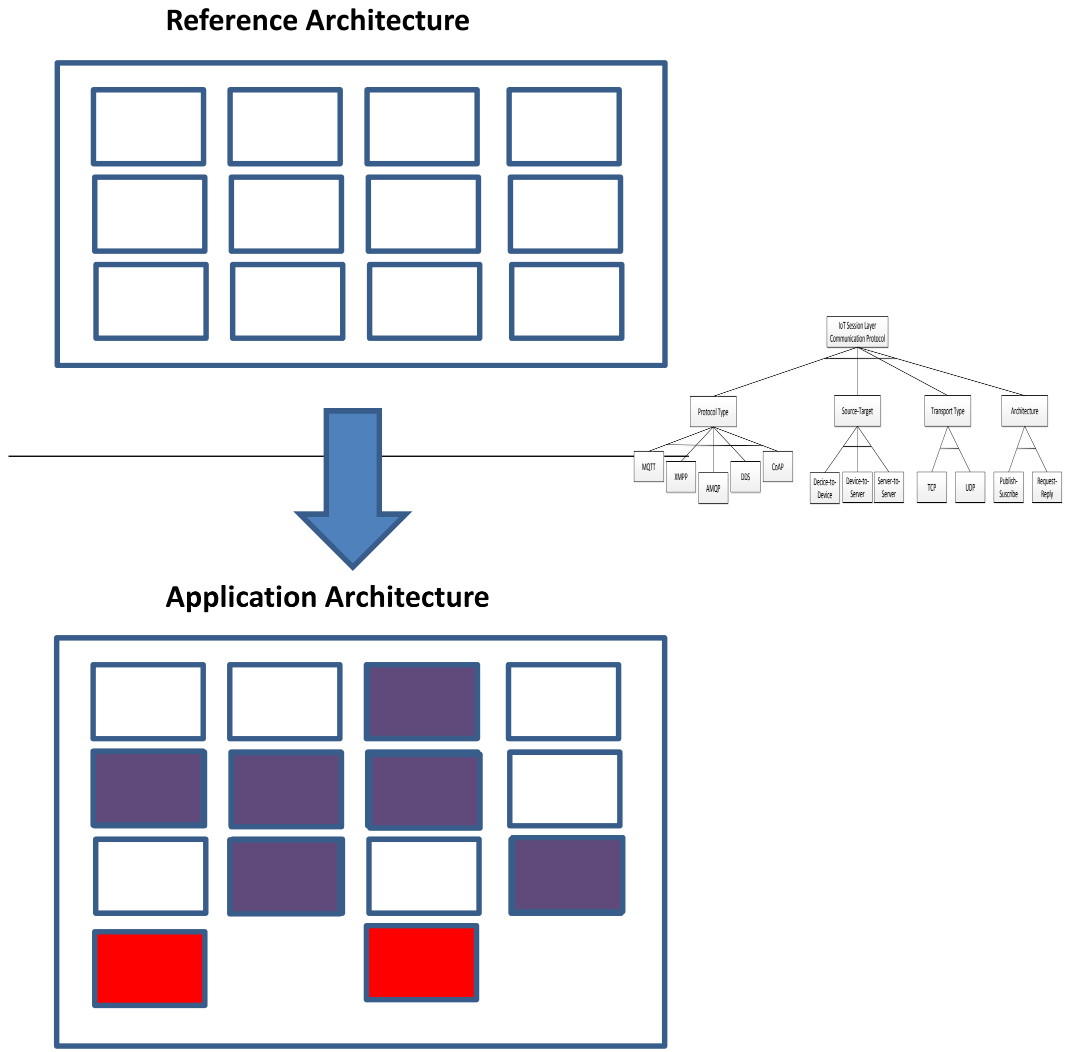


Fig. 9 IoT based FMIS—deployment view

IoT Application Architecture

- Produce the **application architecture** which is a **specialization of the reference architecture** developed
- Application architecture selects the elements of reference architecture
- and if necessary introduces application specific changes according to the application requirement specification



Case Study

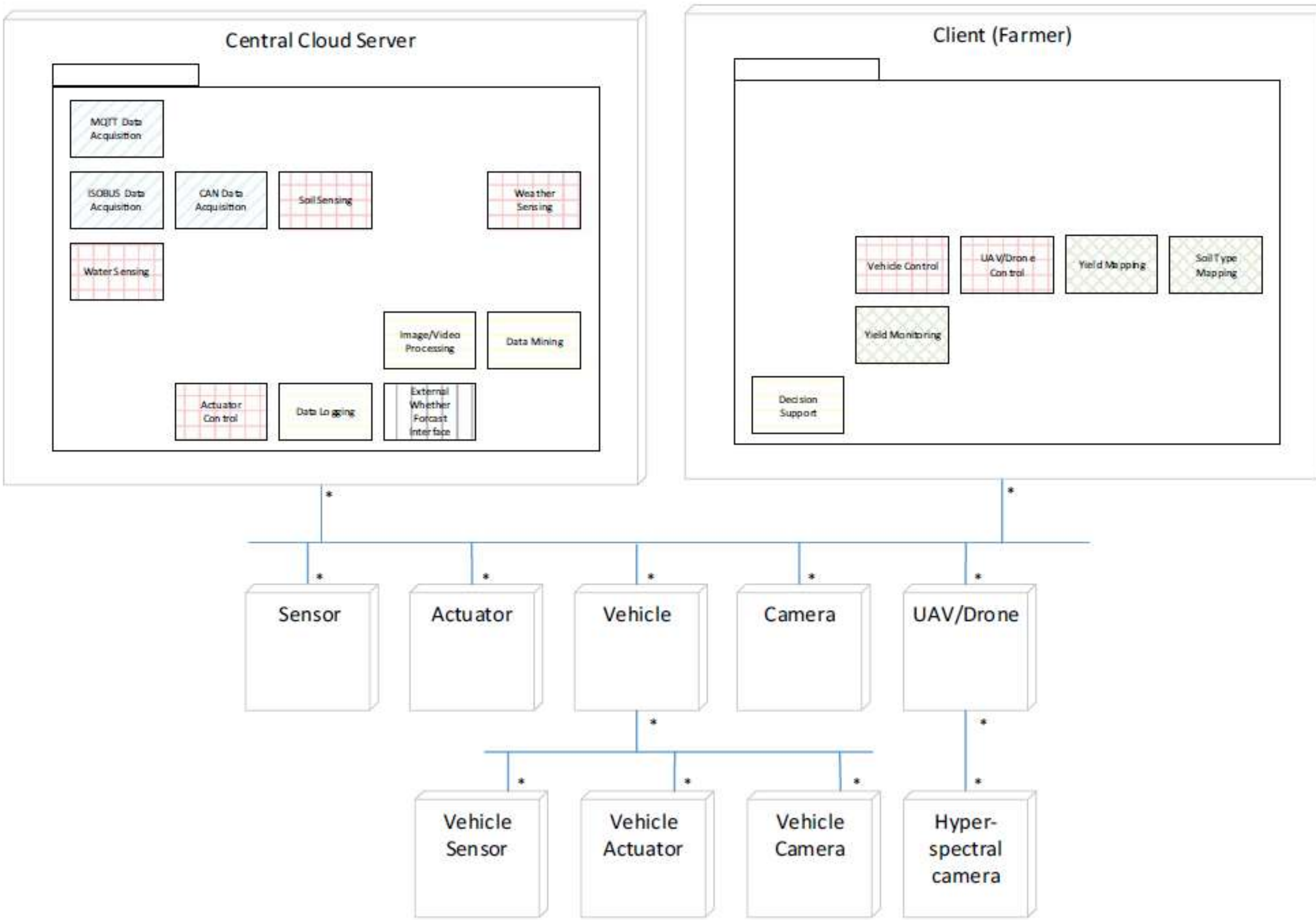


Fig. 13 IoT based FMIS—deployment view of Iot based smart wheat production—retrospective case study

Conclusion

- References architectures for IoT
  - have been provided by different studies
- Yet, the **concrete steps to derive the application architectures are not provided** or implicit
- Commonality and variability analysis, reference architectures and the overall **product line engineering approach can be adopted** to cope with the variability in scope

Precision Agriculture (2019) 20:926–958
https://doi.org/10.1007/s11119-018-09624-8

Architecture design approach for IoT-based farm management information systems

Ö. Köksal[1] · B. Tekinerdogan[1]

Published online: 11 December 2018

Abstract
Smart farming adopts advanced technology and the corresponding principles to increase the amount of production and economic returns, often also with the goal to reduce the impact on the environment. One of the key elements of smart farming is the farm management information systems (FMISs) that supports the automation of data acquisition and processing, monitoring, planning, decision making, documenting, and managing the farm operations. An increased number of FMISs now adopt internet of things (IoT) technology to further optimize the targeted business goals. Obviously IoT systems in agriculture typically have different functional and quality requirements such as choice of communication protocols, the data processing capacity, the security level, safety level, and time performance. For developing an IoT-based FMIS, it is important to design the proper architecture that meets the corresponding requirements. To guide the architect in designing the IoT based farm management information system that meets the business objectives a systematic approach is provided. To this end a design-driven research approach is adopted in which feature-driven domain analysis is used to model the various smart farming requirements. Further, based on a FMIS and IoT reference architectures the steps and the modeling approaches for designing IoT-based FMIS architectures are described. The approach is illustrated using two case studies on smart farming in Turkey, one for smart wheat production in Konya, and the other for smart green houses in Antalya.

Keywords Smart farming · Farm management information system · Internet of things · Architecture design

Feature-Driven Domain Analysis of Session Layer Protocols of Internet of Things

Ömer Köksal
Wageningen University
Information Technology
Wageningen, The Netherlands

omer.koksal@wur.nl

Bedir Tekinerdogan
Wageningen University
Information Technology
Wageningen, The Netherlands

bedir.tekinerdogan@wur.nl

Abstract— **The Internet of Things (IoT) architecture is defined as a layered structure in which each layer represents a coherent set of services. For supporting the communication among the different IoT entities many different communication protocols are now available in practice. For practitioners, it is often not clear which communication protocol is suitable for the various conditions in which the IoT systems need to be operated. In this paper, we focus on the session layer which is responsible for setting up and taking down of the association between the IoT connection points. We adopt a feature-driven domain analysis whereby we define the common and variant features that are related to communication protocols in the session layer. Based on the resulting feature diagram we explicitly characterize the existing session layer communication protocols. Further we define the criteria for selecting the identified communication protocols for the different conditions.**

Keywords: Internet of Things, Session Layer Protocols, Message Queuing Telemetry Protocol, Extensible Messaging and Presence Protocol, Data Distribution Service, Advanced Message Queuing Protocol and Constrained Application Platform.

different protocols for the same purpose. From this perspective IoT can be defined as concourse of devices connected by communication software using different communication protocols. Every layer of the IoT architecture includes its own set of possible communication protocols.

Currently there are dozens of communication protocols that are defined by various different organizations and vendors. For practitioners, it is often not clear which communication protocol is suitable for the various conditions in which the IoT systems need to be operated.

In this paper, we focus on the session layer which is responsible for setting up and taking down of the association between the IoT connection points. The session layer provides services related issues of the session such as initiation, maintenance, and disconnection. As such, frequency and duration of various types of sessions are related with the session layer. Also, session information might enforce encryption and other security measures.

Selection of the session layer protocol depends on many factors such as data size, number of devices to be

THANK YOU