*Jannicke Baalsrud Hauge and Wajid Khilji*

# Smart Production Logistics Concepts

## 1 Didactical Preface

### *1.1 Overview of Didactical Fundamentals*

| | |
|---|---|
| Effort in lecture hours (given by the teacher and facilitator) | Theory: 90 min (if on site 2 classes, if remote 3x30)<br>Safety introduction 45 min on site, 30 min remote<br>Lab operation 60<br>30 min of debriefing after class |
| Effort Self-study (flipped classroom) | Preparation 10h incl. replay<br>10h for report analysis |
| ECTS | 1 ECTS |
| Prerequisites | None, this is an intro course, but basic knowledge on actuators and sensors is an advantage |
| Additional information | To do the experiment, passing the safety test is required<br>A facilitator and a technician need to be on-site in the lab<br>The students should submit lab reports afterwards |
| Adaptability | The usage of the digital twin is flexible. We have a different set of theory if a class is on an advanced level |

### *1.2 Keywords*

Internet of Things, CPS components, SoA, Barriers, CPS application in production logistics

### *1.3 Learning objectives*

- Understand the system architecture
- Know the different components of a CPS system
- Understand how CPS and its components can be applied to improve different aspects of logistics
- Know about the main barriers to implementation
- Know relevant frameworks for standardization in CPS

### *1.4 Target Group*

The course is designed to introduce students at the undergraduate level to cyber-physical applications in production logistics as a part of a smart production concept. It is designed for students in sustainable product development, but it is also usable for students of business administration and for vocational training at different levels. We also use the same digital twin but a different chapter for the master's students in sustainable product development.

## 2 Use Case

### *2.1 User Story*

The cost of logistics is often high, but with a minimal value-added contribution [3,4], and consequently, stakeholders are looking into how to use technology to offer the same or better services at a lower cost. The focus of this unit is to use digital twins either as an on-site lab or as a remote lab. The objective is to investigate how digital twins can contribute to supporting the decision-making process of selecting the right components for a specific company both in relation to the degree of automation and the digitalization of the operations. A summary of the lab environment is given below.

A core component is connecting equipment and RTLS, since these are a prerequisite for being able to realize transparent and efficient material flow from a warehouse to a production site. The initial demonstration scenario is to kit three different parts into two different kits by introducing a UR collaborative robot, an AGV, a Kinect vision system, and an RTLS. The AGV transports the parts from their storage position to the UR robot station, which performs the kitting with the help of the vision system. The digital twin is used both with pure simulation data as well as with data from what we already have in operation, which is currently mostly related to the integration and interaction of the AGV and picking robot, investigating how the AGV is doing its mission scheduling, and how this can be improved. A main action for the latter is to compare the simulated optimized route in the digital twin to the one the AGV actually chose in the physical environment, and investigate how to best realize the part of the physical environment for the quality checkpoints and how to embed this best in the material flow.

In order to investigate the problems mentioned in section 2, the physical objects and the digital twin are integrated into a single system, whose components communicate via a data streaming bus. This system interconnects all the components to make use of the digital twin in a real-world scenario. As a first step, communication between each component was realized. A data streaming bus, which will hold and send messages wherever needed was chosen for this purpose. Secondly, there is an application layer, which is required in order to write business logic or calculations. In addition, data storage and the digital twin ought to be connected through the application layer. Figure 1 shows the system architecture for the production logistics digital twin.

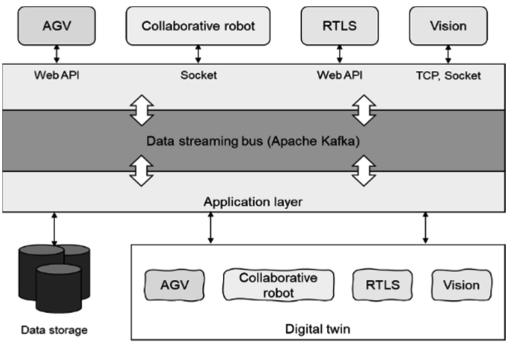



*Fig.1. Production logistics digital twin system architecture*

In the digital environment, replica components of the physical environment are defined. Each replicated component has to go through the data streaming bus and application layer to send and receive information from the physical environment since the operating systems, data formats, and communication methods are different in each component. The next paragraphs describe communication between the components in more detail: A) The AGV communicates via web API. The results of this API are not only location data, but also various parameters, such as battery status, speed, and orientation of the AGV, which may deliver additional valuable information for logistics operations. Of specific relevance is to know that the AGV uses its own coordinates system. This is translated into the digital twin model to align it with the physical environment. B) The collaborative robot uses a pre-defined language to program and communicate with the external world. In this case, the robot communicates via a TCP socket connection, in which the robot is treated as a server and connected via a TCP client application. The robot sends six angular rotations to the unity model (digital twin), for simulation. C) The RTLS is used to identify various locations of the components in an indoor location system. The RTLS system uses API to retrieve location data on these various objects. D) The vision system is currently built by using Microsoft Xbox Kinect. The vision system is currently working via a C# program with the help of an OpenCV wrapper, called EMGU. So far, we have been using Haar's classifier method for deep learning mechanisms to identify an object. The vision system elaborates coordinates from a recognized image, which Xbox Kinect contains in properties of an image object that have X, Y, and Z coordinates. Coordinates will be sent to a middleware that translates them into the collaborative robot coordinates. E) To visualize the digital twin, Unity is used. Along with some complex prefabs like the AGV and collaborative robot, delivered by the supplier or modeled using IPS Each controller is responsible for fetching data or controlling the machine. The digital twin can visualize the track of the AGV to analyze deviations. F) Data storage is used for the long-term data solution for data analytics or data mining. This will help to create a way to replay a particular scenario. Message brokers can be used instead, but later there will be a need to store data for a longer period of time. This layer is important to unify the data transport layer. It provides an opportunity for replication. In addition, there will be a single point of data disposal and retrieval, which makes it easier for many other devices that need data from a single point.

More information can be found in the referenced articles under additional material.

### *2.2 Tasks*

Tasks for students:

- Read the chapter below
- Open the virtual lab and explore the different components (we can use inquiry-based, problem-based, or project-based learning here)
- Watch the two different videos on material handling—identify 10 changes in terms of changes
  - Operational processes
  - Safety issues
  - Efficiency
  - Personal integrity
  - Quality of operation

– Discuss with your co-students the changes you observed. How will this affect future needs in terms of
  - Working environment?
  - Skills development in the workforce?
  - Internal organization?

Self-test

– Define the components of the system to be deployed (e.g., sensors, devices, actuators, storage entities, mobile app);
– Define the functionalities of each component
  - e.g., sensors read data related to different operations, the devices collect data from sensors and forward them to the central entity and/or process them to control actuators
– Outline the data flow among the different components
  - e.g., sensor data goes from IoT nodes to the central entity, from the central entity to the mobile app.

Exercise

– Remote Lab exercise
– Deliver lab report

### *2.3 Learning Resources*

- Script (see below)
- A set of articles
- Example articles:
- Baalsrud Hauge, J.; Zafarzadeh, M.; Jeong, Y.; Li, Y.; Ali Khilji, W.; Larsen, C.; Wiktorsson, M (2021) Digital twin testbed and practical applications in production logistics with real-time location data
- Barreto et al. Indusry4.0 implication in logistics; https://ac.els-cdn.com/S2351978917306807/1-s2.0-S2351978917306807-main.pdf?_tid=f5a8ca13-498d-4064-94f2-b945083ca1d4&acdnat=1553020335_a994bf025db102fd694707c668a1ca97
- H.Deramy: Architectural Design Principles For Industrial Internet of Things, Ch. 2–3 http://www.diva-portal.org/smash/get/diva2:1199862/FULLTEXT01.pdf
- K.L.Goedecke: Development of a Cyber-Physical Logistics System in an Industrie 4.0 Environment, Chapter 4 ;https://www.iwi.uni-hannover.de

/fileadmin/wirtschaftsinformatik/Abschlussarbeiten/K_MA_Goedecke.pdf

- Zafarzadeh, M., Baalsrud Hauge, J., Wiktorsson, M. (2019) Real-time data gathering in production logistics: A research review on applications and technologies affecting environmental and social sustainability; Full Paper Template (diva-portal.org)

## 3 Introduction: Smart Production Logistics

The term Industry 4.0 aggregately alludes to a wide scope of current ideas, whose reasonable order concerning control as well as their exact qualification is unbelievable in singular cases. In the following, accompanying basic ideas are recorded [1]

**Smart Logistics:** In the following five to 10 years, smart supply chains will exert a solid pull on the worldwide economy. The change in perspective is already in progress, as linear and sequential supply chain activities move to a digitized, open framework—part of Industry 4.0—that boosts machine abilities with high throughput and limited resources. [1]

**Cyber-physical Systems:** The physical and the computerized level blend. In the event that this covers the level of generation as well as that of the products, frameworks rise whose physical and computerized representation cannot be separated in a sensible way any longer. A case can be observed within the area of preventive upkeep: Handle parameters (stretch, beneficial time, etc.) of mechanical components and basic (physical) wear and tear are recorded carefully. The genuine condition of the framework comes about from the physical object and its computerized process parameters. [1]

**Self-organization:** Existing manufacturing frameworks are becoming progressively decentralized. This comes together with a deterioration of classic generation progression and a shift towards decentralized self-organization. [1]

**New systems in distribution and procurement:** Dispersion and acquirement will be progressively individualized. Associated forms will be taken care of by utilizing diverse channels. [1]

**New systems in the development of products and services:** Product and service improvement will be individualized. In this setting, the approaches of open development and item insights as well as item memory are of exceptional significance. [1]

**Adaptation to human needs:** New manufacturing systems should be designed to meet human needs instead of the reverse. [1]

**Corporate Social Responsibility:** Sustainability and resource efficiency are continuously at the center of the plans of mechanical manufacturing processes. These variables are principal system conditions for successful products. [1]

## 4 ILO 1: Know the different components in a CPS

This course is embedded in a bigger context. The table below gives an overview of what we have in the different levels for different types of studies. The one in this chapter is bachelor-level engineering.

| Topic | Bachelor's | Master's |
|---|---|---|
| Components of CPS system | Engineering, Business | Engineering Business |
| Computer and its different components | Engineering, Business | |
| Embedded Systems | | Engineering, Business |
| Enterprise Software Systems | | Engineering, Business |
| Real-time Information Processing | | Engineering, Business |
| Software design | Engineering, Business | Engineering, Business |
| Modularization | | Engineering |
| Software development | | Engineering |

### *4.1 Components of a CPS System*

A cyber physical system is comprised of cyber components and physical components. That is why it is known as a cyber physical system. A CPS is based on information processing in computer systems, which is embedded into many products, like a car, planes, microwave ovens, or other devices. These computer systems are used to perform specialized tasks.

For example, in a plane, an embedded system would be the autopilot system to control the trajectory of the aircraft or set the altitude, speed, and direction. This computer system interacts with the outside physical environment with the help of sensors and actuators. These embedded systems are no longer standalone; they share their data via communication networks such as cloud data Warehouses via the internet, where data from many embedded systems can be collected and processed, thus creating a system of systems. Connected embedded systems, moreover, can be controlled and decentralized by a computational unit.


*Embedded Systems [2]*

#### 4.1.1 Examples of physical Components

1. Actuators (Maybe can add images?) elbows of a robot—use videos
2. Sensors—accelerometer maybe—and usage.
3. Computer system—videos
4. Embedded System—videos
5. Visioning system—different types of them

**Actuator**: The pressure driven actuator consists of a barrel or liquid engine that uses hydraulic control to encourage mechanical operations.

https://www.youtube.com/watch?v=5CkJGRsoBcs&ab_channel=Firgelli-Automations

TASK: What is the function of air in an actuator?

**Sensor**: A sensor is a device that measures physical input from its environment and changes it into information that can be translated by either a human or a machine.

https://www.youtube.com/watch?v=5b5xJu8KYrc&ab_channel=RealPars

TASK: What are the 4 industrial revolutions?

**Computer system**: A computer system is a set of integrated devices and peripherals that is used in many modern offices and industries. Its functions are to input, output, process, and store information and data.

**Embedded system**: A type of computer system that is installed in many electronic devices, such as microwaves, aircrafts, and cars.

**Visioning System:** Vision systems offer assistance; collaborative robots perform assignments such as assessing, recognizing, tallying, measuring, or reading barcodes. Ultra-high-speed imaging and focal point quality encourage multi-operations in one process.

https://www.youtube.com/watch?v=3Re8EV3jQ6o&ab_channel=CognexTV

### 4.1.2 Examples of Cyber Components

1. Networks—network diagrams—Add tasks—why they are important?
2. Database—Add tasks
3. Augmented reality (is not the focus in this unit)
4. Virtual reality (is not the focus in this unit)
5. Graphics

**Networks**: A computer network is a group of computers connected with each other. It allows all the computers to communicate with each other and share their assets, information, and applications.

https://www.youtube.com/watch?v=cNwEVYkx2Kk&ab_channel=NetworkDirection

Long video

TASK: What is the difference between a Switch and a Router?

**Database**: A structured set of data stored in a computer system, especially one that is accessible in different ways or locations.

https://www.youtube.com/watch?v=Tk1t3WKK-ZY&ab_channel=LinuxAcademy

TASK: What is the difference between a Relational and a Non-Relational Database?

**Augmented Reality**: This is a technology that is used to superimpose a computer-generated picture onto a user's view of the genuine world, hence giving a composite view.

**Virtual Reality**: a computer-generated environment with scenes and objects that appear to be real, making the user feel they are immersed in their surroundings. The computer-generated recreation of a three-dimensional picture or environment that can connect the user with the computer-generated object, individually using extraordinary electronic hardware, such as a protective cap with a screen interior or gloves fitted with sensors.

**Computer Graphics**: Computer Graphics include technology to present visual representations of graphical data. The Method transforms and presents data in a visual shape. Computer design has become a common component in client interfacing, TV commercials, and motion pictures. Graphical representation can be done using a web, desktop, or mobile applications, or by building a digital twin system using gaming engines.

## *4.2 Embedded Systems*

As its title suggests, embedded in an embedded system implies something that is connected to another thing. An embedded system can be thought of as a computer equipment framework with a computer program embedded in it. An embedded system can be an autonomous system, or it can be a portion of a huge system. An embedded system may be a microcontroller or chip-based system, which is outlined to perform a particular task. To illustrate, a fire alert is an embedded system: it will immediately sense smoke, but it is limited to sensing smoke only.

A typical embedded system consists of three components:

- Hardware.
- Application software.
- Real-Time Operating system (RTOS), which is an operating system specifically designed to perform specific tasks. It lets the processor perform processes as per scheduling by following a chain of commands to control latency.

Apparently, we can describe an embedded system as a Microcontroller-centered, software driven, and reliable real-time control system.

### 4.2.1 Characteristics of an Embedded System

**Single-functioned**—An embedded system more often than not performs a specialized operation and does the same thing more than once, i.e., a pager continuously functions as a pager.

**Tightly constrained**—All computing systems have imperatives on design metrics, but those in an embedded system can be particularly tight. Design metrics could be the degree of a system's highlights such as its cost, measure, power, and execution. They must be able to fit on a single chip, must perform quickly enough to prepare information in real time, and devour least power to extend battery life.

**Reactive and Real time**—Numerous embedded systems must persistently respond to changes within the system's environment and must compute certain results in real time without any delay. Consider the case of a car cruise controller; it persistently screens and responds to speed and brake sensors. It must compute increasing speed or decelerations more than once inside a restricted time; a deferred computation can result in failure to control the car.

**Microprocessors based**—This embedded system must contain a microprocessor or microcontroller.

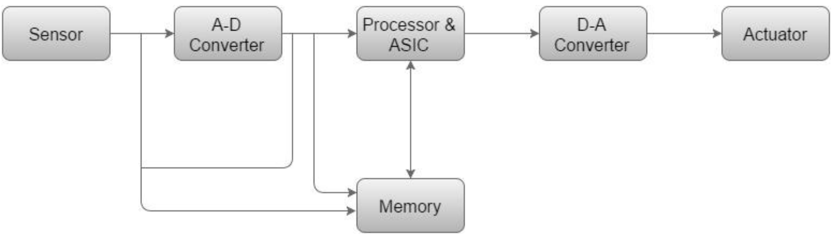
**Memory**—It must have a memory, as its computer program more often than not inserts ROM. It does not need any auxiliary memories within the computer.

**Connected**—It must have associated peripherals to associate input and output devices.

**HW-SW systems**—Program is utilized for more highlights and adaptability. Equipment is utilized for performance and security.

### 4.2.2 Basic Structure of an Embedded System

The diagram below depicts the basic structure of an embedded system:




*Embedded System [8]*

**Sensor**—It measures the physical amount and changes it into an electrical signal which can be studied by a spectator or by any electronic instrument like an A2D converter. A sensor stores the amount measured in the memory.

**A-D Converter**—An analog-to-digital converter changes the analog signal sent by the sensor into a computerized signal.

**Processor & ASICs**—Processors handle the information to measure the output and store it in the memory.

**D-A Converter**—A digital-to-analog converter changes the computerized information bolstered by the processor into analog data.

**Actuator**—An actuator compares the output given by the D-A Converter to the real (anticipated) output stored in it and stores the affirmed output.

## *4.3 Real-Time Information Processing*

A real-time information processing system is able to take input of rapidly changing information and after that give output almost immediately so that change over time is seen promptly in such a system. To illustrate, a radar system depends on a nonstop stream of input information which is prepared by a computer to uncover the number of various aircraft flying inside the extended area covered by the radar and after that show it on a screen so that anybody looking at the screen can know the real location of an airplane at that moment.

Real-time information handling is additionally called stream processing because of the nonstop stream of input information required to abdicate output for that minute. Good examples are online bookings and reservations, e-commerce order processing, and credit card real-time fraud detection. The greatest advantage of real-time information processing is immediate results from input information, which guarantees everything is up to date. Batch handling, on the other hand, implies that information is now not up to date.

#### 4.3.1 Digital Twin System as an Example of Real-Time Information Processing

A digital twin could be a virtual demonstration of a process, product, or service. This blending of the virtual and physical worlds permits information to be investigated and systems to be checked to head off issues before they even happen, anticipate downtime, develop new opportunities, and even plan for the longer term by utilizing simulations. It requires precise data exchange between different devices to work consistently. With the development of 5G, it will be conceivable to work in true real time.

### *4.4 Software Design*

Software design is a process used to convert client prerequisites into an appropriate frame, which enables the software engineer to program, code, and implement software. To survey client prerequisites, an SRS (Software Requirement Specification) document is compiled, while as far as coding and execution are concerned, more particular and point-by-point prerequisites are needed in computer programming terms. The output of this process can straightforwardly be utilized in programming languages. Software design is the primary step in an SDLC (Software Design Life Cycle), which moves the concentration from issue space to arrangement space. It tries to indicate how to fulfill the necessities specified in SRS.

### 4.4.1 Software Design Levels

Software design yields three levels of results:

**Architectural Design**—The architectural design is the most noteworthy theoretical form of the system. It recognizes the software as a system with numerous components in connection with each other. At this level, the creators thought of a proposed arrangement domain.

**High-level Design**—The high-level design breaks the 'single entity-multiple component' concept of structural design into the less-abstracted view of subsystems and modules and delineates their interaction with each other. High-level design centers on how the system's components can be executed in the form of modules. It identifies the measured arrangement of each subsystem and their linking and interaction among each other.

**Detailed Design**—Detailed design bargains with the implementation portion of what is seen as a system and its subsystems within the previous two designs. Detailed design is more detailed concerning modules and their execution. It characterizes the logical structure of each module and their interfaces to link with other modules.

## *4.5 Modularization*

Modularization could be a strategy to isolate a code into numerous discrete and independent modules, which are anticipated to be competent to carry out task(s) freely. These modules may work as fundamental constructs for the complete computer program. Creators tend to design modules in such a way that they can be executed and/or compiled independently. Modular design inadvertently follows the rules of the 'divide and conquer' problem-solving methodology, which is often since there are numerous other benefits connected with the modular design of software. [10]

### 4.5.1 Advantages of modularization:

- Smaller components are easier to maintain
- Program can be divided based on functional aspects
- Desired level of abstraction can be included in the program
- Components with high cohesion can be reused
- Concurrent execution can be made possible
- Desirable from in terms of security

#### 4.5.2 Concurrency

All computer programs are designed to be executed consecutively. By consecutive execution we mean that the coded instructions will be executed one after another, inferring that one part of a program can be activated at any given time. Say, a software program has different modules, then only one of all of its modules can be active at any time of execution. In software design, concurrency is actualized by dividing the computer program into different autonomous units of execution, like modules, and executing them in parallel. In other words, concurrency enables the program to execute more than one portion of code in parallel with each other. It is essential for the software engineers and architects to recognize those modules which can be executed in parallel. [10]

#### 4.5.3 Example

The spell check feature in a word processor like MS Word is an individual module of software that runs together with the word processor itself.

### *4.6 Software Development*

Software development is the process software engineers utilize to build computer programs. The process, also known as the Software Development Life Cycle (SDLC), incorporates a few stages that provide a strategy for building items that meet specialized specifications and client prerequisites.

The SDLC provides a universal standard that software companies can utilize to construct and improve their computer programs. It offers a characterized structure for development groups to follow during the design, creation, and support of high-quality programs. The point of the IT software development process is to construct viable products inside a defined budget and timeline.

#### 4.6.1 Key steps in the software development process

There are six major steps in the software development life cycle, including:

1. Needs identification
2. Requirement analysis
3. Design
4. Development and implementation
5. Testing
6. Deployment and maintenance

#### 4.6.1.1 Needs Identification

Needs Identification could be a market investigation and brainstorming stage in this phase. Before a firm builds a program, it must conduct a broad showcase inquiry about how to determine the product's practicality. Engineers must identify the functions and services the program ought to provide so that its targeted buyers get the most out of it and find it vital and valuable. There are a few ways to induce this data, using input from potential and existing clients and surveys.

#### 4.6.1.2 Requirement Analysis

In this stage, stakeholders concur on the technical and client prerequisites and details of the proposed product to realize its objectives. This stage gives a nitty gritty diagram of each component, the tasks, and of engineers and testing parameters to convey a quality product.

The requirement analysis process includes developers, clients, testers, project directors, and quality assurance. This is often also the stage where software engineers select the program development approach such as the waterfall or V model. The team records the result of this stage in a Software Requirement Specification document, which teams can continuously counsel during usage.

#### 4.6.1.3 Design

Design is the third stage of the SDLC. Partners will talk about variables such as risk levels, group composition, pertinent technologies, time, budget, project impediments, strategy, and engineering design. The Design Specification Document (DSD) indicates the engineering design, components, communication, front-end representation, and user flows of the product. This step gives a format for designers and analyzers and decreases the chances of imperfections and delays in the finished product.

#### 4.6.1.4 Development and Implementation

Another stage is the development and implementation of the design parameters. The developers' code based on the product's specifications and prerequisites coincides with the previous stages. Following company methods and rules, front-end engineers construct interfacing and back ends, whereas database administrators create significant information within the database. The software engineers, moreover, test and survey each other's code. Once the coding is complete, developers send the product to an environment within the implementation stage. This permits them to test a pilot version of the program to create performance and coordinate the necessities.

#### 4.6.1.5 Testing

The testing stage checks the program for bugs and confirms its performance before delivery to clients. In this stage, expert testers confirm the product's capacities to make sure it performs according to the requirements analysis document. Testers utilize exploratory testing in case they encounter a computer program or a test script in order to approve the execution of individual components of the program. They notify engineers of defects within the code. On the off chance that designers affirm the imperfections are valid, they will improve the program, and the testers will rehash the method until the computer program is free of bugs and carries on meeting the desired requirements.

##### 4.6.1.5.1 Deployment and Maintenance

Once the computer program is defect-free, the engineers can supply it to clients. After the release of a type of software's production version, the IT program development company creates a support group to oversee issues clients experience when utilizing the product. Support can be useful if there is a minor issue, but serious computer program failures require an overhaul.

### 4.6.2 Types of software

Software fit into three main clusters based on their usage and application. The popular categories of software are below.

#### 4.6.2.1 System software

Also known as an operating system or OS, system software is the program your computer uses to decipher input commands into machine-readable language. The operating system controls a computer's equipment. Examples of prevalent operating systems utilized in individual computers include the Windows OS from Microsoft, Mac OS utilized in Apple MacBooks, and the Linux-based Ubuntu. Web servers utilize the Apache OS, whereas the UNIX operating system is used to construct proprietary systems.

#### 4.6.2.2 Application software

This type of applications is used by the general public to perform tasks on their computers and smartphones. Well-liked examples are word processing apps, web browsers, media players, picture editing tools, anti-virus software, and even software-as-service (SAS) products.

#### 4.6.2.3 Programming languages

This is the programing language used to produce software systems. It is used solely by coders to make programs. Programming languages include Java, C++, PHP, and Simlab.

## 5 ILO 2: Understand the system architecture

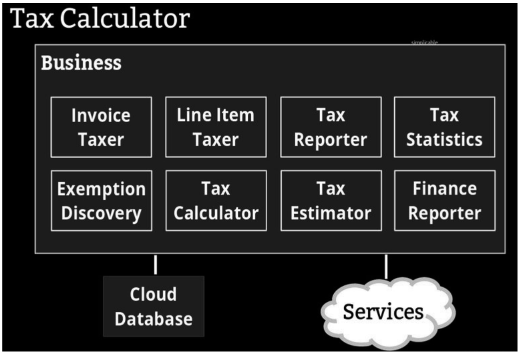
| Topic | Bachelor's | Master's |
|---|---|---|
| Components of the system architecture | Engineering, Business | Engineering, Business |
| Basic Architecture types | | Engineering, Business |
| Advanced Architecture types | | Engineering |

### *5.1 Components of the System Architecture*

System architecture is the structural design of systems. Systems are a class of software that provide foundational service and automation.

#### 5.1.1 Components

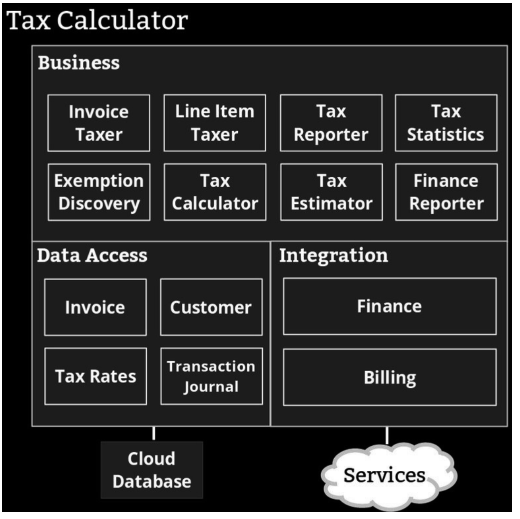
A basic approach to the design is to separate work into components, which are also designed to be reusable. Parts conjointly serve to scale back extraordinarily complicated issues into small, manageable issues. The distinction between a pricey, unstable, low-performance system and a quick, low-cost, and reliable system typically comes all the way down to how well it has been architected into components. For instance, a service for calculating tax for a commerce company may need the following parts. [11]




*Components [11]*

### 5.1.2 Layers

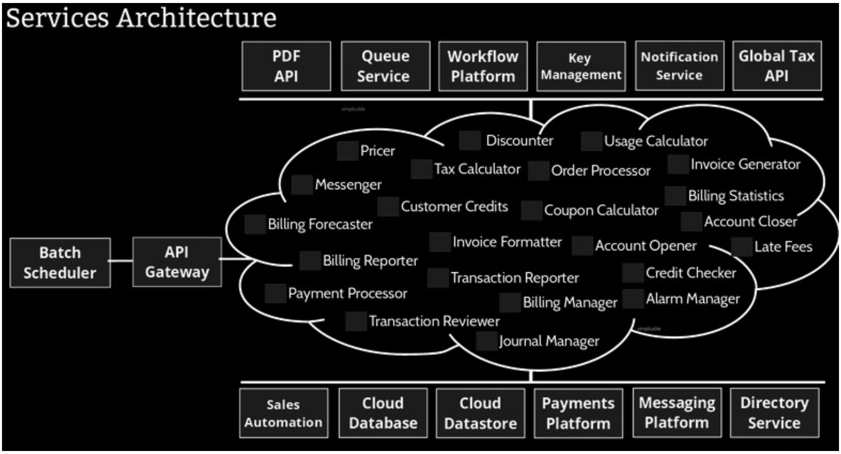
It is common to separate parts into layers. Individual components in numerous layers are loosely coupled so that their implementation is hidden behind an interface. This enables complexity reduction and may cut back the cost of future changes. For example, if a business layer is aware of nothing of how data is stored, then you will be able to change your information without any changes to your business layer.




*Layers [11]*

### 5.1.3 Services

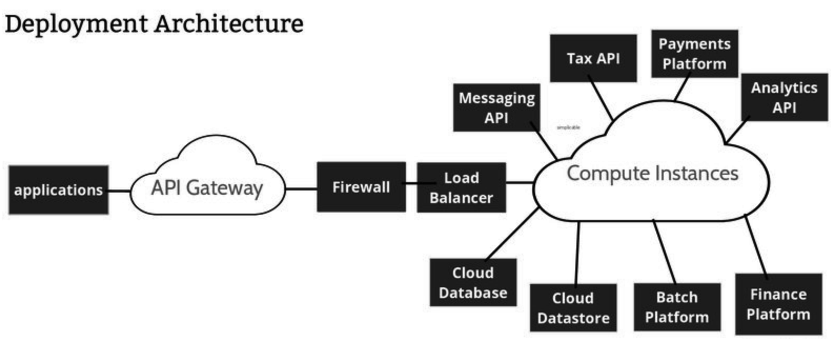
A service could be a piece of functionality that may be deployed and managed on an individual basis. Services are loosely coupled so that you will be able to work on a service while not impacting the remainder of your architecture. For instance, the Tax Calculator service (above) may be one among dozens of services in a billing system.




*Services [11]*

### 5.1.4 Deployment

As services are individually deployed, they permit extreme scalability and dependability. Services can even cut your computing costs as they permit massive systems to be deployed to several instances of cheap hardware.




*Deployment [11]*

In order to build and understand a CPS system, it is important to understand the software architecture or system architecture of CPS. In software engineering, there are many examples of software architecture. We will discuss some of the common software architecture patterns, which are also important for CPS.

## *5.2 Basic Architecture Types*

- Layered (n-tier) Architecture
- Event-bus Architecture
- Microservices Architecture (SoA)
- Client-server Architecture

### 5.2.1 Layered (n-tier) Architecture

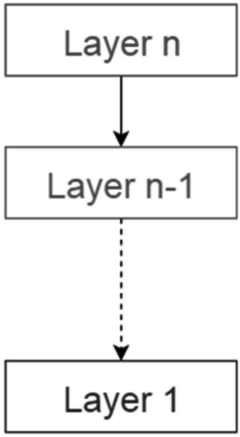
This software architecture pattern creates different levels of abstraction. Each architecture layer provides a service to the next higher-level layer.

The most commonly found four layers of general information are as follows:

- Presentation Layer (AKA UI layer)
- Application Layer (AKA service layer)
- Business logic Layer (AKA domain layer)
- Data access Layer (AKA persistence layer)

Usage:

- Desktop Applications
- E-commerce web applications



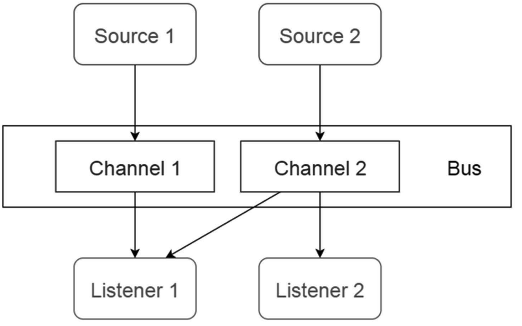

*Layered (n-tier) Architecture [12]*

### 5.2.2 Event-bus Architecture

This architecture type deals with various events and has four major parts. Event source, event listener, channel, and event bus. Sources publish messages to specific channels on an event bus. Listeners subscribe to specific channels. Listeners are notified of messages that are published on a channel to which they have already subscribed. [12]

Usage:

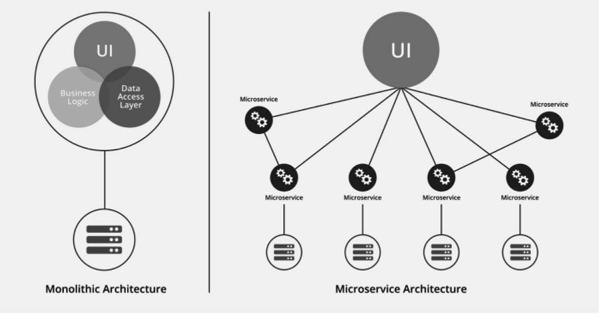
- Android development
- Notification services




*Event-Bus Architecture [12]*

### 5.2.3 Microservices Architecture (SoA)

This architecture has become the most accepted and well known within the past couple of years. It depends on providing tiny, autonomous modular services, where every service solves a selected issue or performs a special task. And these modules communicate with one another through well-defined API to serve the business objective. [13]

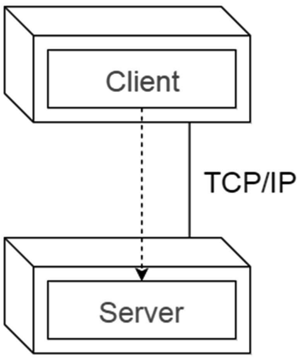



*Microservice Architecture [13]*

#### 5.2.4 Client–Server Architecture

This design comprises 2 parties: a server and various clients. The server component can offer services to numerous client components. Clients ask for services from the server and also the server offers pertinent services to those clients. Besides this, the server proceeds to listen to clients' demands. [12]
Usage:

- Online applications such as email, document sharing, and banking.




*Client–Server Architecture [12]*

### *5.3 Advanced Architecture Types*

- Model-View-Controller Architecture
- Broker Architecture

#### 5.3.1 Model-View-Controller Architecture

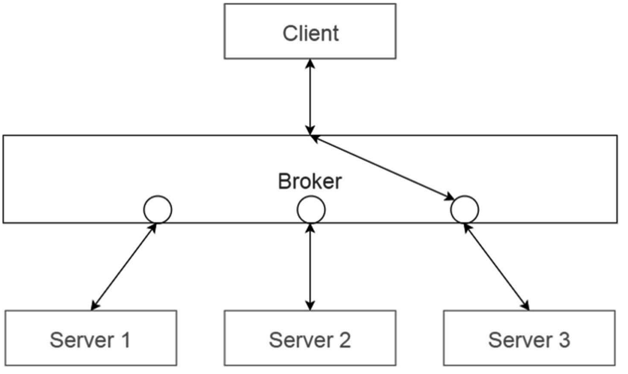
This architecture divides an interactive application into three parts:

- Model—Contains the core functionality of acquiring and storing data.
- View—Contains the functionality of designing and displaying information to the uer.
- Controller—Handles the input from the user

This is often done to isolate inside representations of data from the way data is displayed to and acknowledged by the client. It decouples components and permits proficient code reuse.

Usage:

- Web applications



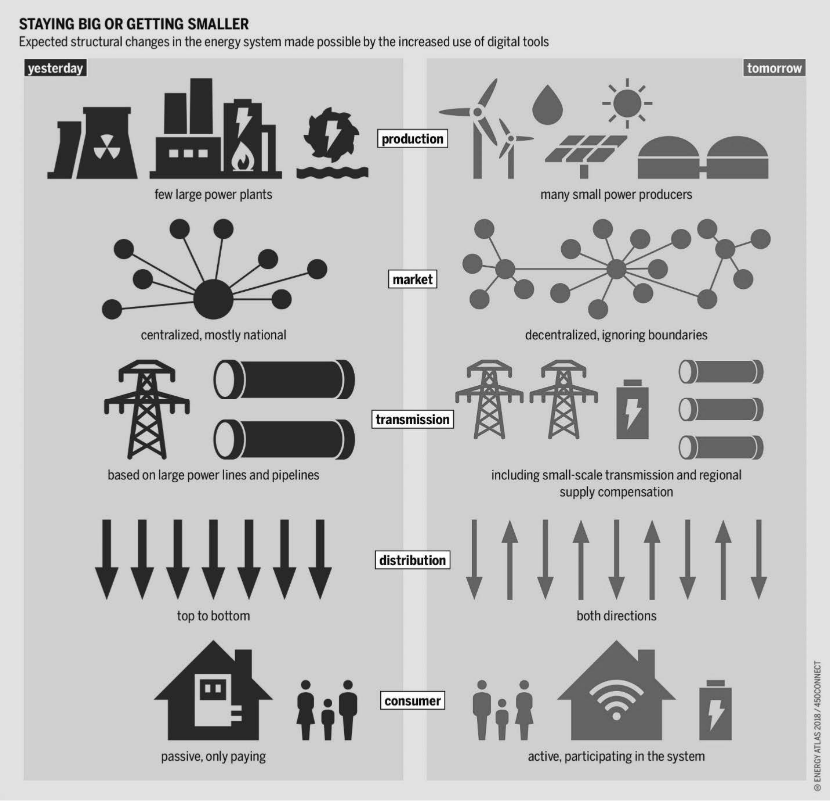

*Model-View-Controller Architecture [12]*

### 5.3.2 Broker Architecture

This design is utilized to structure distributed systems with decoupled components. These components can be connected to each other by remote service invocations. A broker component is usually responsible for the coordination of communication among various components. Servers publish their capabilities (Services and Characteristics) to a broker many times. Clients request a service from the broker, and the broker then directs the client to an appropriate service from its registry. [12]

Usage:

- Message broker software such as Apache ActiveMQ, Apache Kafka, and RabbitMQ




*Broker Architecture [12]*

## 6 ILO 3: Get an overview of how it can be used in logistics — applications

| Topic | Bachelor's | Master's |
|---|---|---|
| Smart Grid | Engineering, Business | |
| Smart Supply Chain Management | Engineering, Business | |
| Autonomous Guided Vehicle | Engineering, Business | |
| Healthcare Logistics | | Engineering, Business |
| Automatic Pilot in Avionics | | Engineering |

In CPS, using physical systems and connecting them via a cyber-system requires knowledge of various domains, such as Information technology, computer networks, software engineering, and robotics. There are various ways in which a logistics system can benefit from CPS. Some common ones are mentioned below: examples related to logistics

- Smart Grid (smart power consumption and distribution)
- Smart Supply Chain Management
- Autonomous automobiles

### *6.1 Smart Grid*

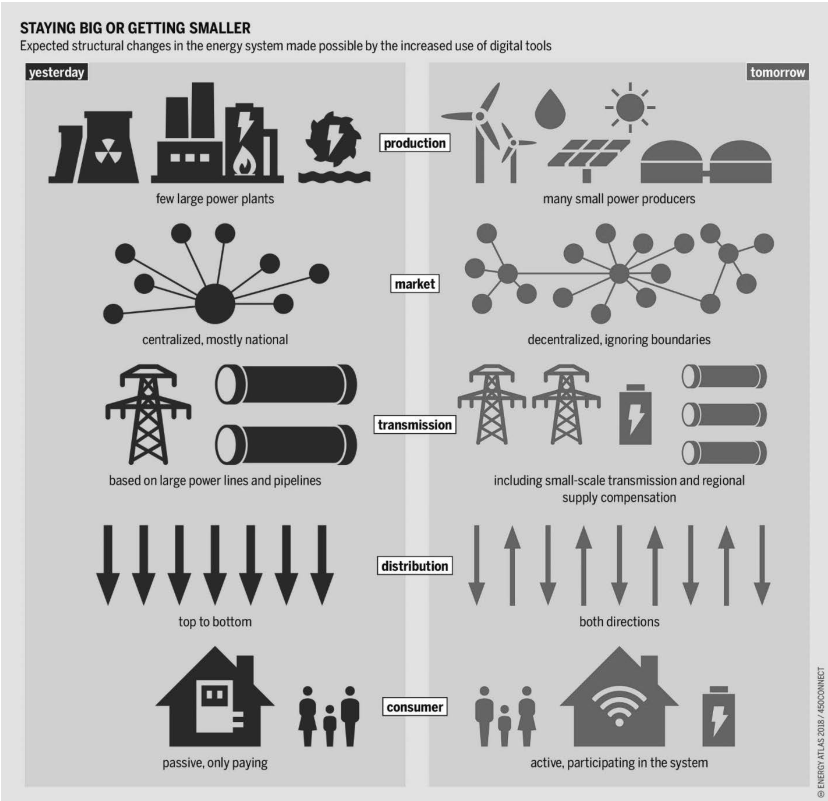
A smart grid is an electrical grid which has a range of operation and energy measures together with smart appliances, smart meters, renewable energy resources, and economical energy resources. [14]

https://www.youtube.com/watch?v=JwRTpWZReJk&ab_channel=U.S.DepartmentofEnergy

TASK: What is Home Area Network?

### *6.2 Smart Supply Chain Management*

The scope of supply chain management is vast—it has the potential to reinforce each supply chain function, from inventory prediction to demand and supply management. Applied properly, smart supply chain management may revolutionize strategic decision-making, creating a really agile and optimized ecosystem. Even simply an increase of a tenth in potency is well worth the effort, netting an organization the maximum amount of $276 billion in fifteen years. Plus, analysis has shown that about 80% of corporations with leading supply chain operations achieve above-average revenue growth. [15]




*Smart Grid [14]*

## *6.3 Autonomous Automobiles*

An autonomous vehicle may be a vehicle which will drive itself with no input from a human driver. These kinds of vehicles are called self-driving cars, driverless cars, or robotic cars.

Computer-controlled and wheel-based, automatic radio-controlled vehicles (AGVs) are load carriers that follow the ground of a facility without an operator or driver on board. Their movement is directed by a mix of software systems and sensor-based steering systems. Because they travel a certain path with precisely controlled acceleration and deceleration, and embody automatic obstacle detection bumpers, AGVs offer safe movement of loads. Typical AGV applications are the transportation of raw materials, work-in-process, and finished goods in support of creating production lines, and storage/retrieval or alternative movement in support of selecting in warehousing and distribution applications. [16]

There are several types of AGVs. These include:

- **Automated carts**—These are the simplest kind of AGV with the least features for the lowest-cost implementation.
- **Unit load AGVs**—This is the kind of individual vehicle that transports loads (typically bins, pallets, carts, or bundles) on forks or on the AGV's deck. Roll-handling AGVs explicitly handle heavyweight rolls of steel or paper.
- **Tugger AGVs**—This type of powered units can pull a series of non-motorized trailers that each carry a load.
- **Automated forklift AGVs**—The type of existing forklift trucks, whose controls have been altered to allow pilotless operation

Typically, battery-powered AGV systems incorporate multiple vehicles that navigate pre-defined paths. Vehicles navigate within the facility using many steering technologies together with floor–surface mounted magnetic tape or bars, lasers, optical sensors, and magnets/gyroscopes primarily based on inertial navigation. These steering technologies make it simple to alter the routes and expand the AGV system in response to facility changes for a versatile and scalable material handling solution. [16]

For real-time control and observation of multiple AGVs, computer-based software systems use wireless connections to gather information concerning every unit's current location, and then interface with a software system for destination and routing logic. The software package directs the vehicles by wirelessly communicating specific tasks to the AGVs via radio frequency (RF). Directions include stops, starts, dynamic speed, lifting, lowering, multi-point turns, reverses, diversions from the prescribed path, and interfacing with alternative material-handling instrumentation and systems—both machine-controlled and static.

AGVs are used in a range of areas to support handling and processing throughout a facility:

- **Assembly**: AGV moves products through production processes
- **Kitting**: AGV collects parts for assembly
- **Transportation**: AGV loads and reloads loading pallets and loose parts

- **Staging**: AGV delivers pallets for production processes
- **Warehousing**: AGV moves items from stretch wrappers to docks or storage

**Order picking**: AGV moves ordered items to trailer-loading zone for delivery, and moves a platform for a picker to place designated items on

**Parts/just-in-time (JIT) delivery**: AGV tows trailers carrying parts or materials to consumption zone

**Transfer/shuttle**: AGV transfers loads across high traffic aisles

## 7 ILO 4: Know about the main barriers to implementation

| Topic | Bachelor's | Master's |
| --- | --- | --- |
| Barriers to Smart Manufacturing | Engineering, Business | |
| Barriers to AI adoption | | Master's |

### *7.1 Barriers to Smart Manufacturing*

Technology is not a roadblock to the present endeavor; in fact, the technology is already here. On discussing the barriers to smart manufacturing throughout the session, the audience unanimously agreed that technology was not an element in this respect. They provided the subsequent list of barriers: [19]

**Economic**: Decision makers cannot always see the benefits of actualizing innovation into the manufacturing process. Whereas someecision makers can be paralyzed by the investment and exertion of procuring modern innovation, specialists must see past the starting investment, considering the time and money that will eventually be gained by joining such an environment.

**Social/Cultural**: This is often the more common complaint that encompasses most developments. It is often related to different aspects, like lack of organizational preparedness, reluctancy towards technology implementation and uptake. This happens when there is an uncovered need for direct implementation and execution of new technology from a technical perspective, but appear to be working fine with the status quo to the users or decision makers. One audience member referenced the popular quote from Jim Collins, "Good is the enemy of great," to summarize the acknowledgment of keeping the manufacturing plant in its current state.

**Security**: The expanding utilization of connected devices inside plants and along supply chains causes numerous clients to inquire, "Is our data safe?" Avoidance of security risks begins with quality merchants and analyzing vulnerabilities at each "touchpoint". An audience member referenced a "Hack-a-thon" that their plant had conducted to test zones of weakness inside their connected systems. Once these boundaries were recognized, the discussion at that point turned to a dialog on how they can start a journey toward Smart Manufacturing. [19]

### *7.2. Barriers to AI adoption*

**Fear of AI:** Whereas the fear of terminators invading the office is one take, the genuine fear of AI relates to gigantic work losses and unemployment due to robotization. This is an issue that the world must address incrementally as innovation progresses. The automation being utilized presently is centered on monotonous assignments like mail reminders, information extraction and entry, upgrading spreadsheets, and other administrative chores. Coupled with AI and machine learning, contract supervisors, paralegals, and others get instant information investigation on contracts and other records, which improves productivity and gives them more time to deal straightforwardly with clients. [19]

It is presently essentially inconceivable for administrators and other business leaders to manually oversee all requests, contracts, sellers, compliance, and other obligations. AI and computerization complement today's burdensome workload to grant time back to workers, and rather than doing their occupations for them and making them nonessential, AI makes their workday more effective and compelling. When done right, this eventually leads to more joyful workers and a more advantageous bottom line.

**Mountains of data:** Another obstruction is the enormous amounts of information that companies have collected over the last few decades or more. Where do you start? What can you do with it? How do you keep it secure and compliant with directions? The arrangement here is to not get scared into inaction, as each business can use the information on hand to conduct profitable, predictive analyses. The great news is the more information, the better prepared the AI will be. And as for security, information kept in a single secured system is always more secure than information scattered over a few machines or put away in record cabinets. [19]

**Picking the right algorithm:** Although this is a critical step in creating AI, it is not one that will be made in the starting stages of setting up an AI strategy, unless the company develops their AI completely in house, typically likely a choice to be made by an AI computer program provider. Whether it is an out-of-the-box AI apparatus or a custom-built AI application, businesses can presently actualize AI with as much or as little customization as they need.

**Shortage of data scientists:** This can be a huge problem for the future of AI programs, and there are numerous thoughts on how to unravel the issue. As companies ramp up their efforts to utilize machine learning and AI, data science is becoming a critical component of a successful strategy. But when executing any program, the most vital steps are building up the business's needs which the program will solve and then conveying those needs. Data science plays an imperative part in turning raw information into business value, but current workers that already know the information can be trained, experts can be utilized, and outsourcing this work to sellers is becoming commonplace. [19]

## References

H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," Bus. Inf. Syst. Eng., vol. 6, no. 4, pp. 239–242, Aug. 2014, doi: 10.1007/s12599-014-0334-4.

"Applications of Embedded Systems." https://www.rs-online.com/designspark/applications-of-embedded-systems-1 (accessed Mar. 30, 2021).

P. Kamau, "The Computer Motherboard and Its Components," *Winstar Technologies*, Aug. 17, 2019. https://winstartechnologies.com/the-computer-motherboard-and-its-components/ (accessed Mar. 30, 2021).

"CPU (Central Processing Unit) Definition." https://techterms.com/definition/cpu (accessed Mar. 30, 2021).

"Gaming," *Fromagerie St. Nicholas*. http://www.fromagerie-st-nicolas.com/category/gaming/ (accessed Mar. 30, 2021).

"How to increase RAM and add new memory." https://pbuy2021.gq/ProductDetail.aspx?iid=28283123&pr=88.99 (accessed Mar. 30, 2021).

"SSD vs HDD: Which Do You Need?," *SSD vs HDD: Which Do You Need?* https://www.avast.com/c-ssd-vs-hdd (accessed Apr. 08, 2021).

J. Sankar, "Embedded Systems: An Overview | Basic Electronic Tutorials," *Electronics For You*, Jun. 11, 2017. https://www.electronicsforu.com/resources/embedded-systems-overview (accessed Mar. 30, 2021).

"Enterprise software," *Wikipedia*. Feb. 28, 2021, Accessed: Mar. 30, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Enterprise_software&oldid=1009428713.

"Software Design Basics," p. 3.

"4 Examples of System Architecture," *Simplicable*. https://simplicable.com/new/system-architecture (accessed Mar. 30, 2021).

V. Mallawaarachchi, "10 Common Software Architectural Patterns in a nutshell," *Medium*, Sep. 02, 2020. https://towardsdatascience.com/10-common-software-architectural-patterns-in-a-nutshell-a0b47a1e9013 (accessed Mar. 30, 2021).

iitbitz, "Medium article," *Sharing Knowledge and programming*, Nov. 09, 2018. https://iitbitz.wordpress.com/2018/11/09/medium-article/ (accessed Mar. 30, 2021).

"Smart grids: what is a smart electrical grid — electricity networks in evolution," *i-SCOOP*. https://www.i-scoop.eu/industry-4-0/smart-grids-electrical-grid/ (accessed Mar. 31, 2021).

B. Ballapragada, "Scope for Smart Supply Chain Management in Industry 4.0," *ThroughPut*, Dec. 04, 2019. https://throughput.world/blog/topic/smart-supply-chain-management/ (accessed Mar. 31, 2021).

"Automatic Guided Vehicles." https://www.mhi.org/fundamentals/automatic-guided-vehicles (accessed Mar. 31, 2021).

"Medical Monitoring 101: Providing Medical Expertise for Clinical Trials — C3i Solutions." https://www.c3isolutions.com/blog/medical-monitoring-101/ (accessed Mar. 31, 2021).

Y. Jia, L. Guo, and X. Wang, "4 Real-time control systems," in *Transportation Cyber-Physical Systems*, L. Deka and M. Chowdhury, Eds. Elsevier, 2018, pp. 81–113.

"Smart Manufacturing: What is it? What are the Barriers? Where Do I Begin?," *iBASEt*, May 13, 2016. https://www.ibaset.com/smart-manufacturing-what-is-it-what-are-the-barriers-where-do-i-begin/ (accessed Mar. 31, 2021).

B. Melzer, "Reference Architectural Model Industrie 4.0 (RAMI 4.0)," p. 15.

## Authors

Prof. Dr.-Ing Jannicke Baalsrud Hauge
Bremer Institut fuer Produktion und Logistik (BIBA)
KTH Royal Institute of Technology, Hochschulring 20
28359 Bremen, Germany
Kvarnbergagatan 12, 15181 Södertälje, Sweden
https://www.kth.se/profile/jmbh;
https://www.biba.uni-bremen.de/institut/mitarbeite-rinnen/homepage.html?nick=baa
baa@biba.uni-bremen.de; jmbh@kth.se

Wajid A. Khilji
KTH Royal insitutute of Technology, Kvarnbergagatan 12,
15181 Södertälje, Sweden
https://www.kth.se/profile/wakhilji
wakhilji@kth.se