|  |  |  |
| --- | --- | --- |
| **MODULE 4 of 5** | | **|** |
| Intro to the new ICT Inventory for Education in Emergencies ... | **System Architecture |** | |

**SYSTEMS ARCHITECTURE**

In the previous module, we discussed System Integration. And we learned systems are a class of software that offers automation and foundational functions as opposed to acting as a tool for people to use. Alternatively, they may offer a user interface for purposes such as administration, operation, and management.

There are three sections to be considered:

* [a **unified definition**](https://www.lix.polytechnique.fr/~golden/systems_architecture.html#definition) of Systems Architecture (outlining what are systems, their architecture, and the justifications for Systems Architecture)
* [**fundamental principles**](https://www.lix.polytechnique.fr/~golden/systems_architecture.html#principles) underlying Systems Architecture (in all acceptations of the term)
* [**socio-cognitive aspects**](https://www.lix.polytechnique.fr/~golden/systems_architecture.html#others) of Systems Architecture to consider the reality of men (cognitive limitations) and teams (social behaviors).

**(1) A Unified Definition**

* ***Do we have said "architecture"?***

Systems Architecture is a generic discipline to handle objects (existing or to be created) called "systems", in a way that supports reasoning about the structural properties of these objects.

Depending on the context, Systems Architecture can refer to:

* the architecture of a system, i.e. a *model* to describe/analyze a system
* architecting a system, i.e. a *method* to build the architecture of a system
* a body of knowledge\* for "architecting" systems while meeting business needs, i.e. a *discipline* to master systems design.   
  *\* consisting in: concepts, principles, frameworks, tools, methods, heuristics, practices*

At this point, we can only say that the "architecture of a system" is (similarly to the one of a building) a global model of this system consisting of:

* a structure
* properties (of various elements involved)
* relationships (between various elements)
* behaviors & dynamics
* multiple views of the system (complementary and consistent).
* ***But what is a "complex system"?***

Systems are typically objects designed by men, involving heterogeneous components (e.g. hardware, software, humans) working together to perform a mission. The **complexity** of a system will mainly come from two aspects:

* **Integration of components**: there are many interrelations between a possibly huge number of components, and there are recursive levels of integration
* **Heterogeneity of components**: several specialized fields are involved in the design of a complex system, making it difficult to keep a unified vision of this system and to manage its design.

You can read further explanations of [complex systems](http://www.lix.polytechnique.fr/~golden/system.html). We will not describe here the numerous issues raised (at every level of a company: corporate strategy, marketing, product definition, engineering, manufacturing, operations, support, maintenance, etc.) by the design and management of such complex systems. But every experienced business leader, manager, consultant, or engineer will perfectly know what we are talking about. These issues can be summarized as:

* **going from local to global**, i.e. mastering integration and emergence
* **building an invariable architecture** in a moving environment.
* ***And so, Systems Architecture is ...?***

In this context, **Systems Architecture is a response to the conceptual and practical difficulties of the description and the design of complex systems**. Systems Architecture helps to describe consistently and design efficiently complex systems such as:

* an industrial system (the original meaning of Systems Architecture)
* an IT infrastructure (Enterprise Architecture)
* an organization (Organizational Architecture)
* a business (Business Architecture)
* a project (Project Architecture?).

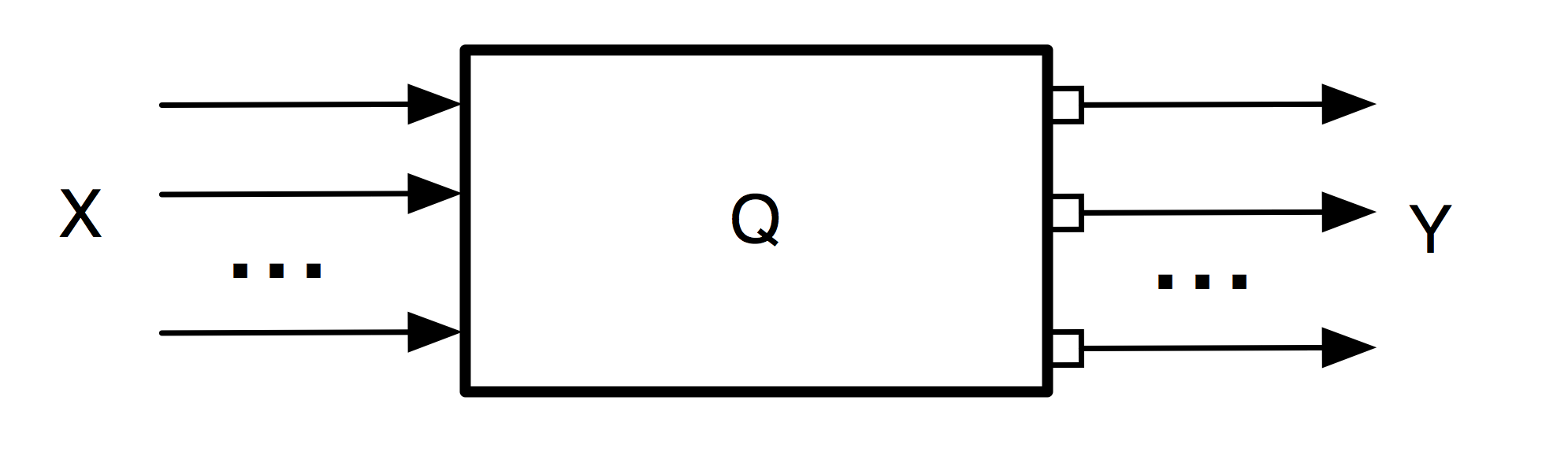
Systems Architecture will often rely on a tool called an ***architecture framework***, i.e. a reference model to organize the various elements of the architecture of a system into complementary and consistent predefined views allowing to cover all the scope of Systems Architecture. Famous architecture frameworks are for example [DoDAF](http://en.wikipedia.org/wiki/Department_of_Defense_Architecture_Framework), [MoDAF](http://en.wikipedia.org/wiki/MODAF), or [AGATE](http://en.wikipedia.org/wiki/AGATE_(architecture_framework)).

Finally, Systems Architecture will **consider any system with a socio-technical approach** (even when dealing with a "purely technical" system). During the design (or transformation) of a system, the systems in the scope of this design (or transformation) can be divided into two separated systems in interaction:

* the **product**, i.e. the system being designed or transformed
* the **project**, i.e. the socio-technical system\* in charge of the design or transformation of the product.   
  *\* teams, tools, other resources and their organization following strategies & methods*

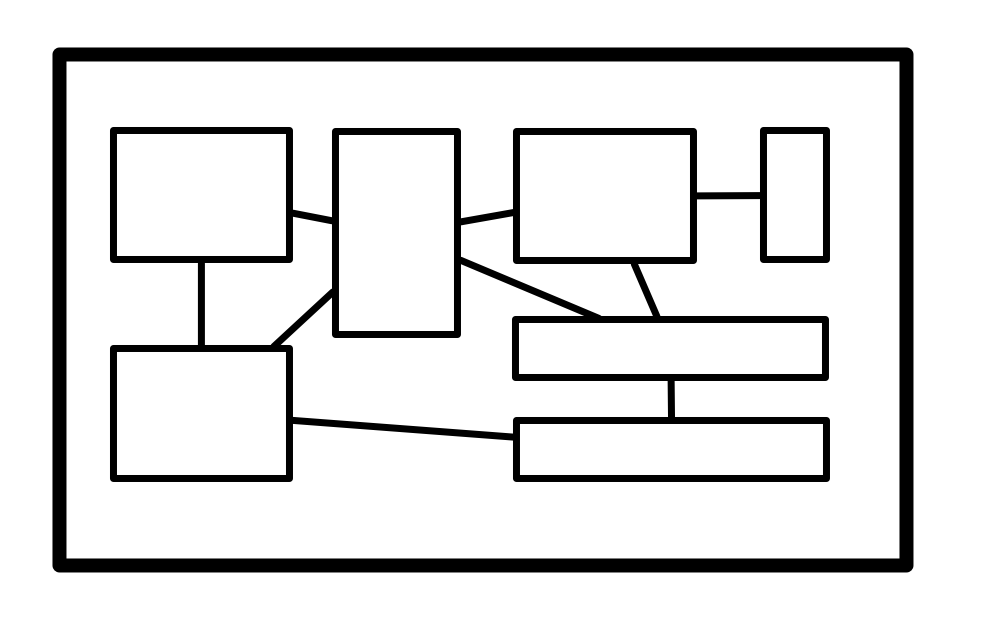
**(2) Fundamental Principles**

Whatever the type of system and the acceptation considered (model, method, or discipline), Systems Architecture is based on nine (9) fundamental principles:

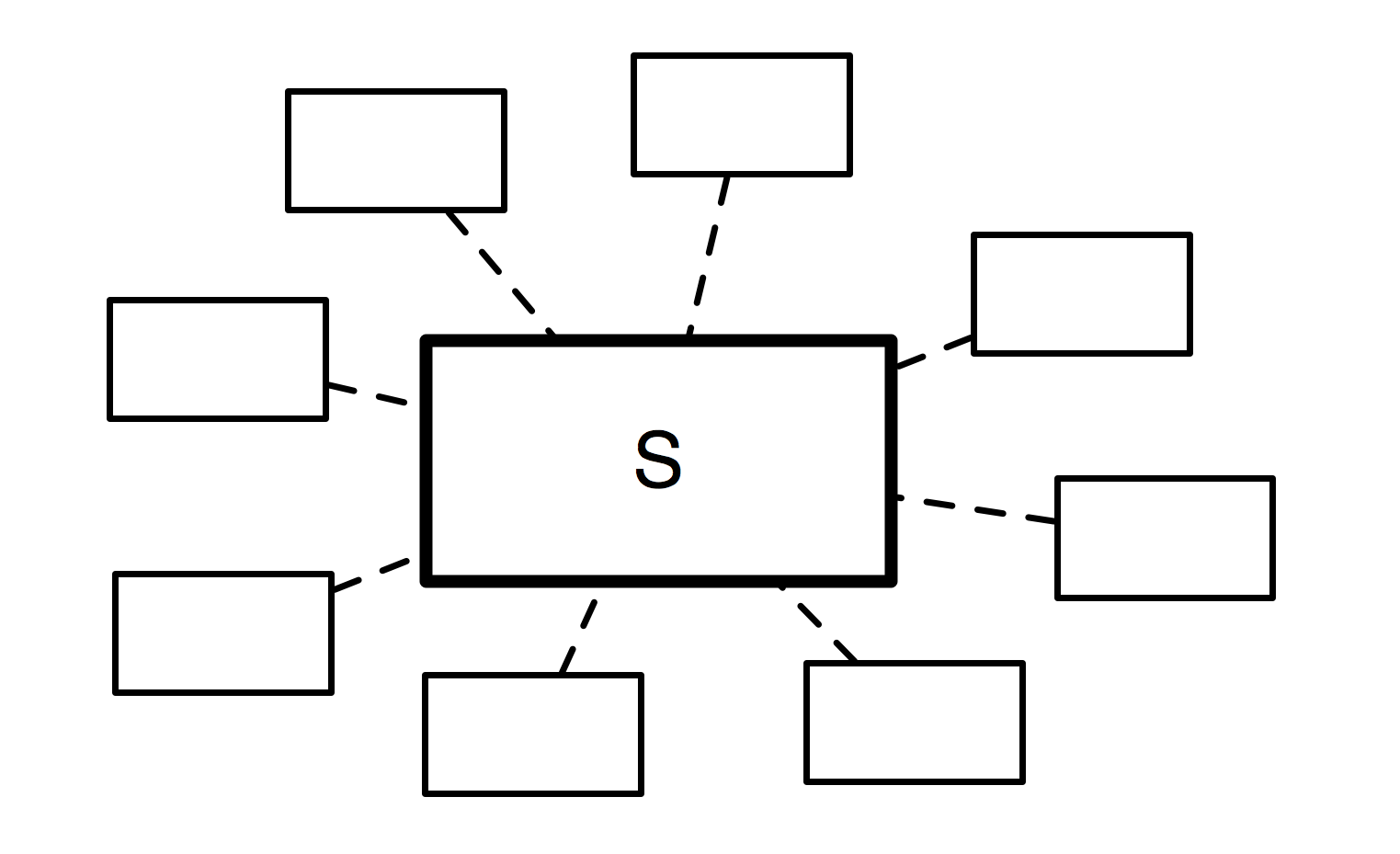
* ***"Thinking with a systemic approach"*** 
  1. **the objects of the reality are modeled as systems** (i.e. a box performing a function and defined by its perimeter, inputs, outputs, and an internal state)

*e.g. A mobile phone is a system that takes in input a voice & keystrokes and outputs voices & displays. Moreover, it can be on, off, or on standby. Overall, the phone allows making phone calls (among other functions).*

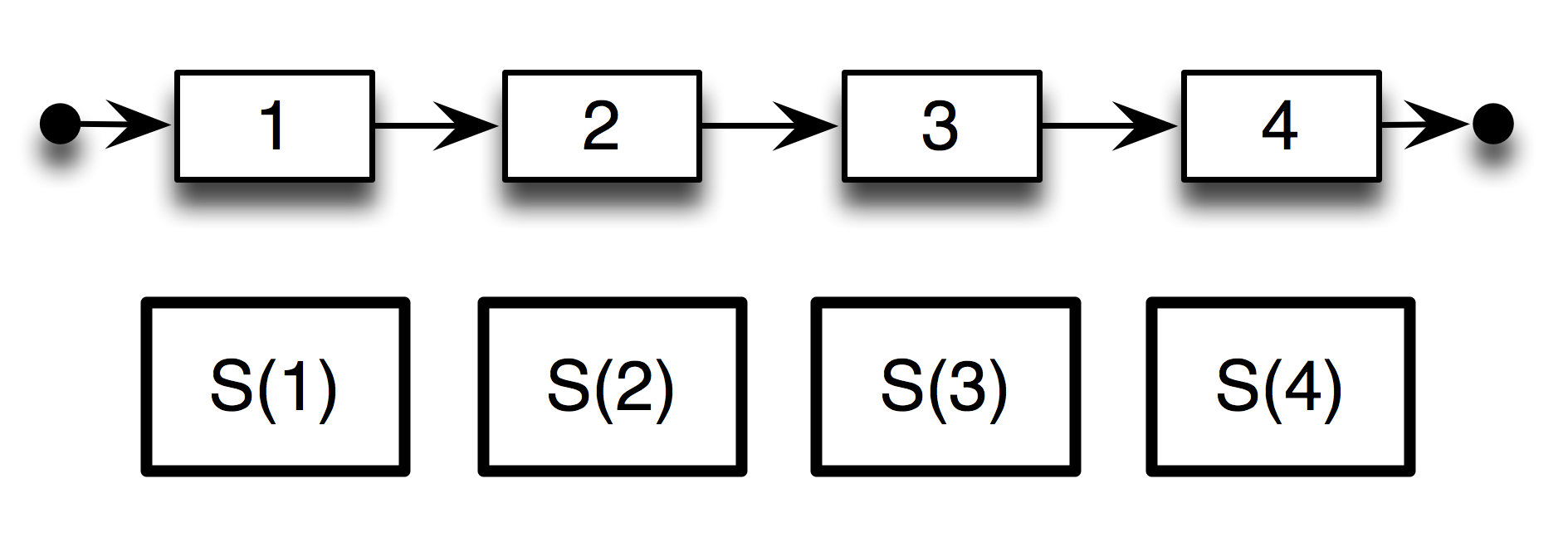
* 1. **a system can be broken down into a set of smaller subsystems**, which is less than the whole system (because of emergence)



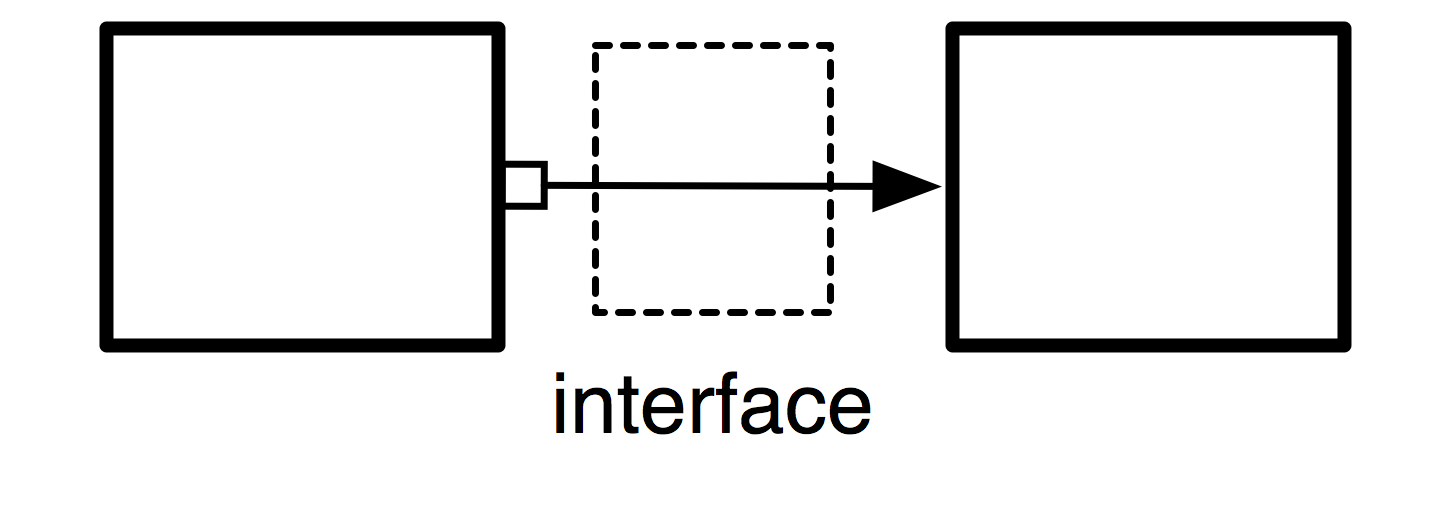
*e.g. A mobile phone is a screen, a keyboard, a body, a microphone, a speaker, and electronics. But the phone is the integration of all those elements and cannot be understood completely from this set of elements.*

* 1. **a system must be considered in interaction with other systems**, i.e. its environment

*e.g. A mobile phone is in interaction with users, relays (to transmit the signal), reparators (when broken), the ground (when falling), etc. All these systems constitute its environment and shall be considered during its design.*

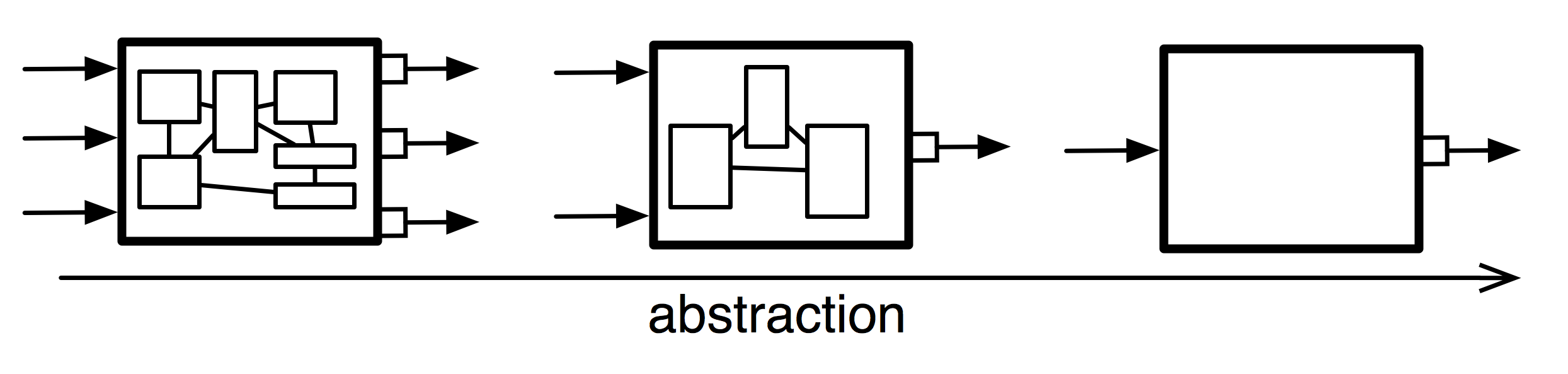
* 1. **a system must be considered through its whole lifecycle**

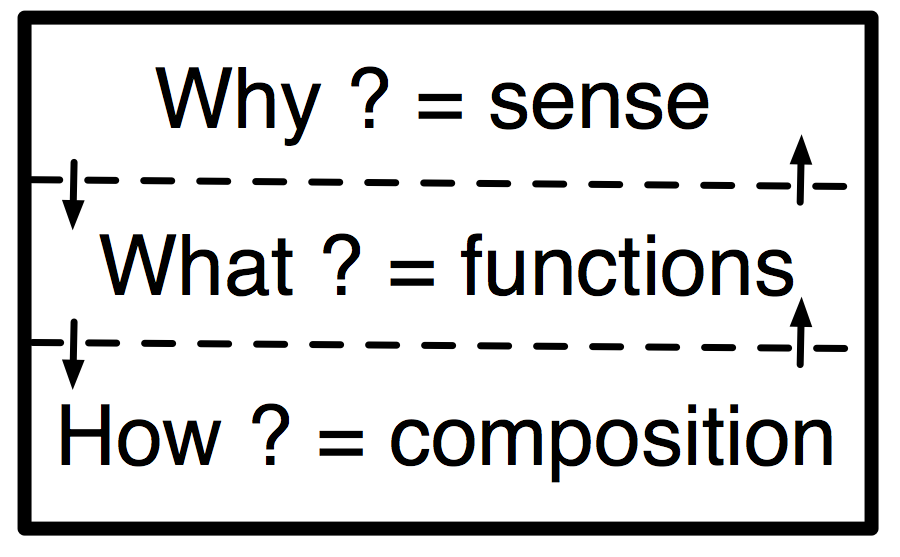
*Ex: a mobile phone will be designed, prototyped, tested, approved, manufactured, distributed, sold, used, repaired, and finally recycled. All these steps are important (and not only the moment when it is used).*

* *"Reasoning according to an architecture paradigm"*
  1. **a system can be linked to another through an interface**, which will model the properties of the link

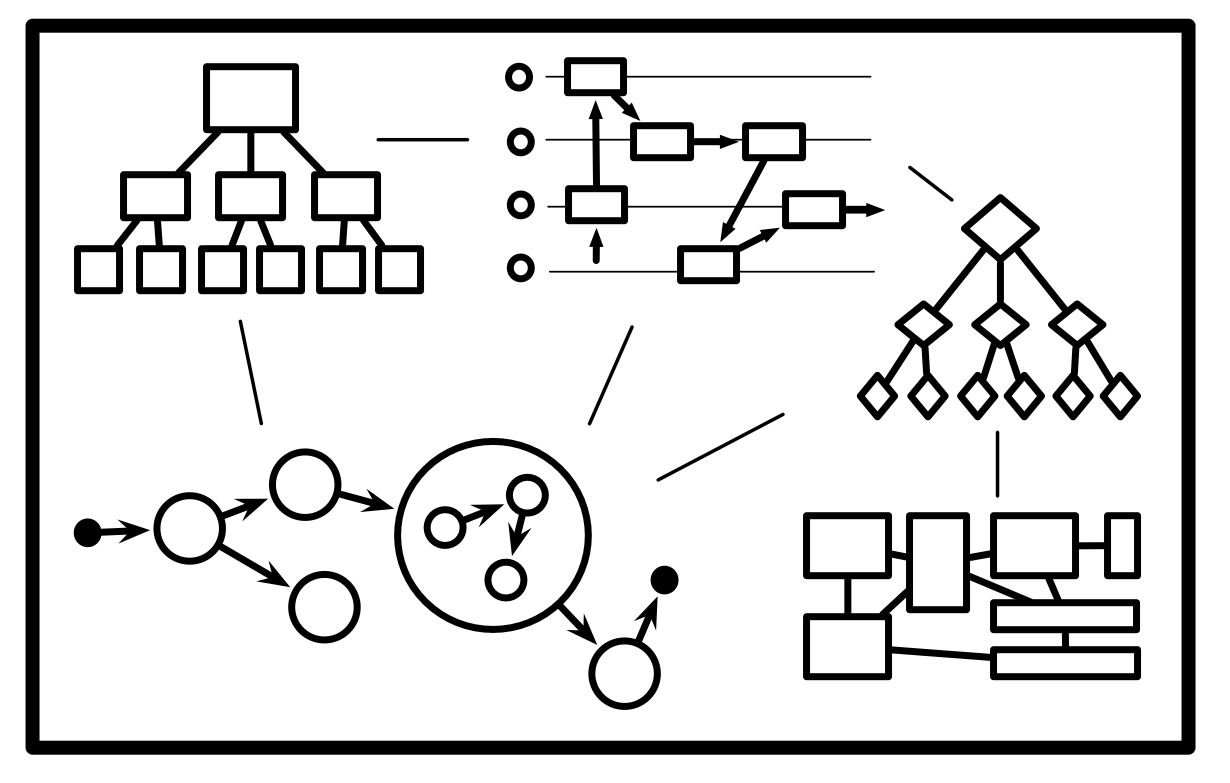
*e.g. When phoning, our ear is in direct contact with the phone, and there is, therefore, a link between the two systems (the ear and the phone). Thus, there is a hidden interface: the air! The properties of the air may influence the link between the ear and the phone (imagine for example if there is a lot of noise).*

* 1. **a system can be considered at various abstraction levels**, allowing to consider only relevant properties and behaviors

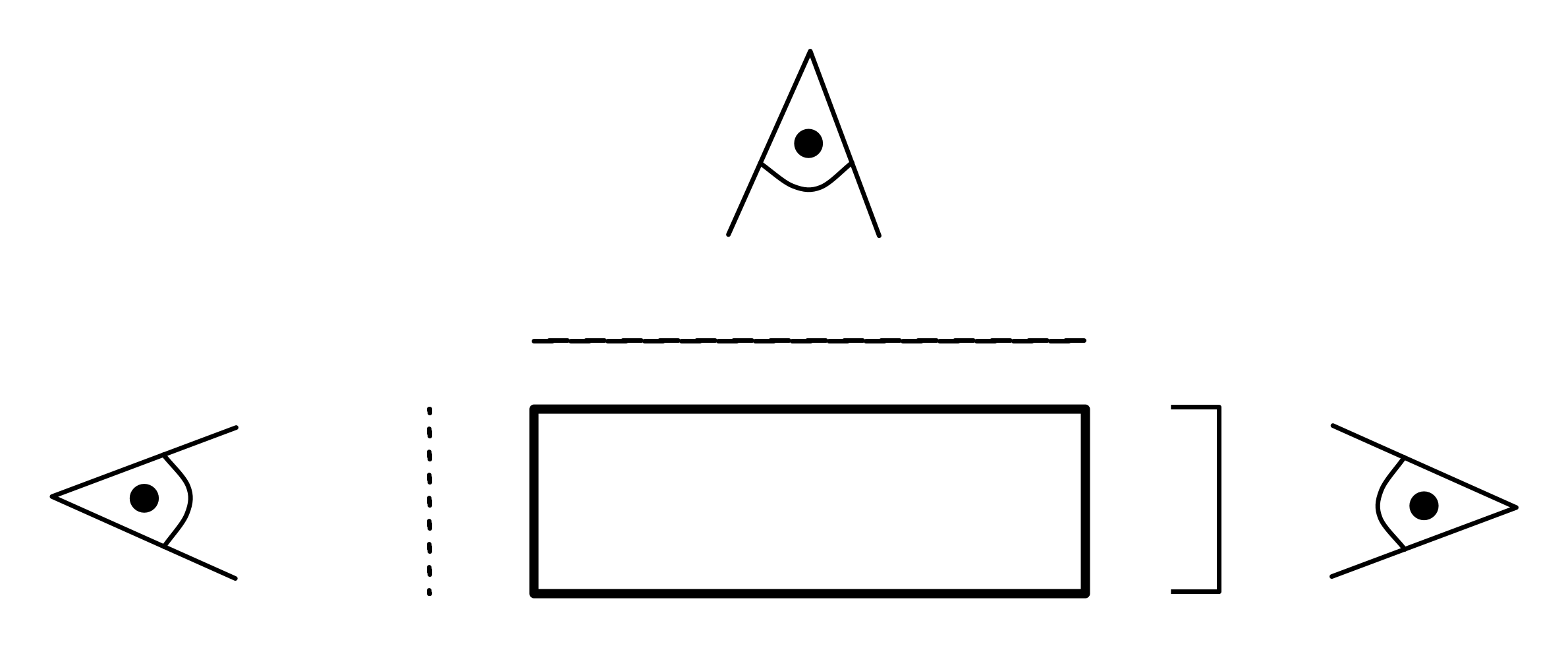
*Ex: Do you consider your phone as a device to make phone calls (and other functions of modern phones), a set of material and electronics components manufactured together, or a huge set of atoms? All these visions are realistic, but they are just at different abstraction levels, whose relevancy will depend on the context.*

* 1. **a system can be viewed according to several layers** (usually three: its sense, its functions, and its composition)

e.g. *A phone is an object whose sense is to accomplish several missions for its environment: making phone calls, being a fashionable object, offering various features of personal digital assistants, etc. But it is also a set of functions organized to accomplish these missions (displaying on the screen, transmitting the signal, delivering power supply, looking for user inputs, making noise if necessary, etc). And finally, all these functions are implemented through physical components organized to perform these functions.*

* 1. **a system can be described through interrelated models** with given semantics (properties, structure, states, behaviors, data, etc.)

*e.g.: From the point of view of properties, the phone is a device expected to meet requirements like "a phone must resist to falls from a height of one meter". But a phone will also change state: when a phone is off and that the power button is pressed, the phone shall turn on. Function dynamics of the phone are also relevant: when receiving a call, the screen will display the name and the speaker will buzz, but if the user presses no button the phone will stop after 30 seconds... This will typically be described with diagrams in SysML (an evolution of UML).*

* 1. **a system can be described through different viewpoints** corresponding to various actors concerned by the system.

*e.g: Commercials, designers, engineers (in charge of the software, electronics, acoustics, materials, etc) users, repairers... All these people will have different visions of the phone. When the designer will see the phone as an easy-to-use object centered on the user, the engineer will see it as a technological device that has to be efficient and robust. A commercial may rather see it as a product that must meet clients' needs and market trends to be sold. All these visions are important and define the system in multiple and complementary ways.*

**(3) Socio-Cognitive Aspects**

Systems Architecture involves multiple views (sometimes partial or conflictual) of the same system by multiple actors. These views can be understood as "projections" of the system in the spaces of those different actors:

* this is the set of all those views (themselves involving several interrelated models at different abstraction levels) that define the system. But it is in general impossible to define a system in an objective, unified and exhaustive way.
* each view is an analytical description of the system. However, the complexity of systems and their architecture cannot be grasped by an analytical decomposition. Considering multiple viewpoints allows compensating the weaknesses of analytical decomposition (which is the only one we can handle as humans), following the ideas of Edgard Morin and Jean-Louis Le Moigne.

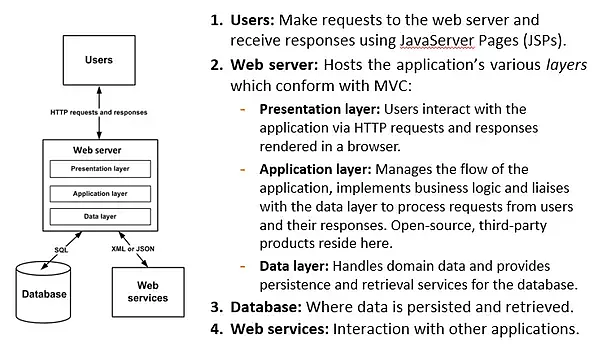
Moreover, the social and cognitive aspects of Systems Architecture are critical to carrying out a successful design. Indeed, Systems Architecture is key to make individual and collective work more efficient in projects, to meet business needs (quality, delays, performances, costs, risk):

* for individuals, Systems Architecture is a powerful tool helping to overcome the complexity of systems and to keep a vision of their work. It allows us to describe, model, and design systems richly and diversely, while keeping good usability of the objects handled and improving decision making. For instance, cognitive rules as *"7x7x7"* (i.e. for one model, there must be at most 7 elements per level and most 3 recursive levels) will allow people to work efficiently with their cognitive limitations.
* for teams, Systems Architecture proposes a common language to understand and be understood. It is also a strong tool & method to facilitate collaboration in projects and to create transversality between departments of a company while allowing to make the right questions emerge in the discussion. In particular, it can help actors to create a shared vision of the system and to converge on various issues. For example, the rule *"Every element of the architecture of a system must have an owner"* will help teams to advance their work without losing the traceability of responsibilities.

Finally, Systems Architecture is not only a model or a method to design complex systems. It is more of a discipline, allowing one to consider at the same time the system and the project in charge of it while overcoming the difficulties related to **the [complexities](https://www.lix.polytechnique.fr/~golden/sociotechnical.html) (technical, social, and cognitive) of the system and its design**.

To wrap up, **System architecture** is the structural design of systems. It is the structural design of software that automates work. Systems are a class of software that provides foundational services and automation. The following are illustrative examples of the system architecture.

Below is a sample system architecture of a website (design or detailed diagram). Since you are the author of your website, this will be your next task.

1. **Users:** Make requests to the webserver and receive responses using JavaServer Pages (JSPs).
2. **Web server**: Hosts the application's various layers which conform with MVC:
   1. **Presentation layer:** Users interact with the application via HTTP requests and responses rendered in a browser.
   2. **Application layer:** Manages the flow of the application, implements business logic, and liaises with the data layer to process requests from users and their responses. Open-source, third-party products reside here.
   3. **Data layer**: Handles domain data and provides persistence and retrieval services for the database.
3. **Database:** Where data is persisted and retrieved.
4. **Web services:** Interaction with other applications.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | **|** In detail, make a ***SIMPLE*** system architecture of your website from the User 🡪 UI 🡪 elements (e.g. HTML, CSS, PHP) or components 🡪 APIs 🡪 and Database. You can make your own architecture design or you can search from the internet for a suitable architecture design for your website, or you may modify those samples presented previously (above). |
|  |  | ***Draw the System Architecture of your Website*** | |
|  |  |  | |
|  |  |

* Good Work!*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| See the source image | | | **| LEARNING SUMMARY** | |  | |
|  |  |  | | *In this module, I learned that…* | |  | |
|  |  |  | |  | |  | |
|  |  |  | |  | |  | |
|  |  |  | |  | |  | |
|  |  |  | |  | |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | *Evident with my…* |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | *In this module, I need to know more about the concepts of…* |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | **| REFLECTION** | |  | | |
|  | | | | | |
|  |  |  | | *Now that you have already familiarised yourself with System Architecture (mobile and website), what did you realize in a real-life situation?* | | |  | |
|  |  |  | |  | | |  | |
|  |  |  | |  | | |  | |
|  |  |  | |  | | |  | |
|  |  |  | |  | | |  | |
|  |  |  | |  | | |  | |
|  |  |  | |  | | |  | |
|  |  |  | |  | | |  | |

* YES! We are done with our* ***4th module****!*

|  |  |
| --- | --- |
|  | **|REFERENCES** |

* *Spacey, 20180, “What is Systems Architecture ?”, retrieved October 20, 2020 from https://simplicable.com/new/system-architecture-definition*
* *Golden, n.d., “What is Systems Architecture?”, retrieved October 18, 2020 from https://www.lix.polytechnique.fr/~golden/systems\_architecture.html*
* *TechPedia, n.d., “What is Middleware? Technology’s Go-to Middleman”, retrieved October 18, 2020 from https://www.techopedia.com/definition/9614/system-integration-si*
* *Lehtonen, 2018, “What is system integration?”, retrieved October 18, 2020 from https://www.youredi.com/blog/what-is-system-integration*