industrial internet CONSORTIUM

I^2M^2–the Future of Industrial System Monetization

Author:
Marcellus Buchheit
President and CEO
Wibu-Systems USA Inc.
mabu@wibu.com

INTRODUCTION

With the evolutional change of industrial systems to connected IIoT systems, many innovations in the industrial world will be possible, such as long-range information exchange, data analytics in the cloud or sophisticated automatic remote control. This article describes another direction: Innovation in the method in which industrial systems will be paid for in the future as these systems and their components will soon be connected to the Internet.

MOTIVATION

Industrial systems are typically large and expensive to build. Most payment is done upfront – after the system is built and ready to use, the operational user takes “ownership” by paying for the entire system upfront – before the system returns any revenue. It could take years before such a system finally turns the first profit defined as return on investment (ROI). All of the financial risk is on the operational user’s side – if the operation of the system permanently fails or is not profitable, there is the risk of high financial loss or even bankruptcy. If the operational user does not have the capital to pay for the system, a loan is required, provided typically by banks or venture capital groups. If the profitability of the planned system is at high risk, banks are usually very restrictive with such loans regardless of the level of innovation or potential advantages for the human community or environment. Venture capital groups are more risk-orientated but their financial capabilities are limited.

As a result, the promise of many highly innovative systems is never realized due to their high risk. Or, in order to build a lower cost system, the builder has to provide a significant discount which is also passed on to the suppliers of the integrated components (devices, software, complex control equipment, etc.). With this one-time payment upfront, there is also no contingent relationship between the system and component builders during the operational phase of the system. As a result, there is no sharing of the operational risks, nor is there sharing of the operational opportunities of the built system either.

How great would it be if the system was paid for based on operational usage rather than the large upfront payment: The operational user would pay for the usage of the system out of the revenue generated by the system. This is different from the traditional leasing system, which still has full upfront payments to the system builder with an upfront loan, which is paid back in fixed intervals by the operational user. Instead, our visionary model would pay the system builder and all the component builders based on usage. This approach strengthens the bond between user and builders who share the risk and opportunities of the operation. For example, the operator of an oil refinery could pay the builders the actual usage costs as part of his revenue, thus relieving the operator of the large upfront investment in the equipment while providing builders with additional risks and opportunities due to a fluctuating oil price.

CHALLENGES AND OPPORTUNITIES

There are two main reasons why this type of payment model does not exist today:

1. A typical large industrial system incorporates many thousands of components (physical parts, devices, software) from many hundreds or even thousands of different builders. There is no efficient manual tracking method possible to deliver usage payments to all these providers. Even traditional automatic payment systems do not really help: Using an individual payment method for each component (e.g. based on credit card and serial numbers as with today's desktop software) would be extremely cumbersome for the operational user.
2. Some of such components are payable upon delivery. This includes physical building parts, electric wiring, mechanical equipment, etc. It does not make sense to pay for such physical components during usage.

The first challenge could be solved with a technical standard, which automatically manages the payment for all components at the operational user site by a centralized cloud-based payment processor (see Figure 1). Much like a credit card clearance service, this processor collects the money one time per payment cycle (e.g. one month) from the operational user and distributes it in accordance with the agreed upon price to all the builders of the components and the system builder (as its share for planning, designing and deploying the complete system).

The second challenge (cost of physical components) will not be fully eliminated in the future but rather weakened: In recent history the cost of physical components is quite stable for most industrial systems. On

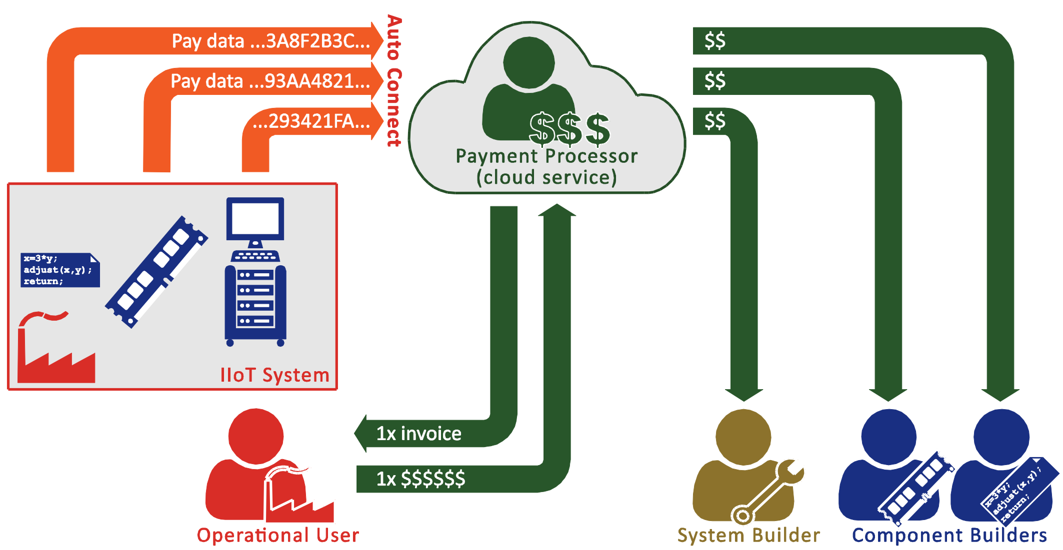

Figure 1: An automatic standardized payment system for industrial systems

the other hand, the cost of 'intellectual' components such as intelligent devices, smart controllers, all types of IIoT devices and software products is continuously rising. The main reason is that the development of such components depends more and more on complex software as well as extensive research whereas the physical parts (e.g. boards, displays and keyboards) are actually getting cheaper due to more usage of standardized parts, resulting in larger production quantities. The cost relationship between hardware and software is shown in Figure 2.

such standardized hardware, more software development is required to adapt the standardized hardware to the specific purpose of a component.

SOFTWARE LEADS THE WAY

The software industry has been moving to a usage-based payment model for several years, typically adopting monthly or yearly subscriptions. Microsoft© prefers the Office 365® subscription payment model over the lifetime payment of the traditional Office Professional package [1]. Another large company, Adobe, sells its Creative Suite only via subscription[2]. In the cloud world, there are absolutely no lifetime licenses as all services are subscription- or pay-per-use-based with the "as-a-service" model.

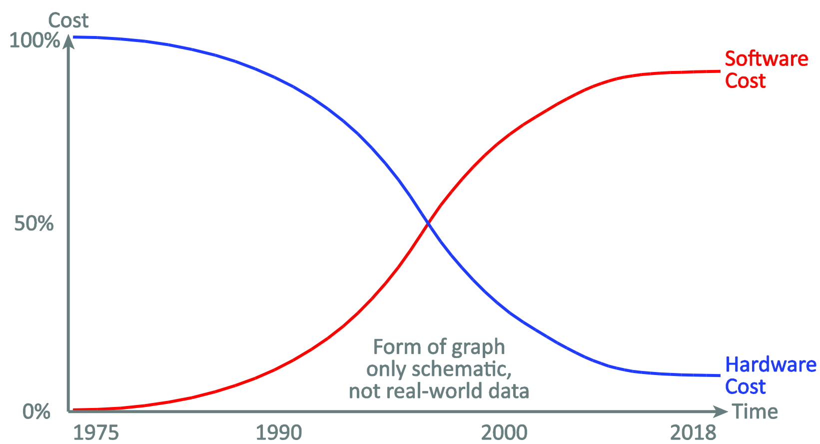

Figure 2: Historic software / hardware relationship in a device

Before 1975, the microprocessor and its software did not exist and the cost was 100% hardware-based. Over time, more and more expensive electronic parts (e.g. a specific analog signal processor or a video controller for the display) were either totally replaced by software, or reduced in price by cheaper standardized mass-produced hardware, such as the Raspberry PI board. But to use

With greater reliance on software in future devices and other internet connected components, the industrial world will be able to discover and monetize the value of

[1] Microsoft Office 365, "personal" is offered for US-$69.99 per year. *https://www.microsoft.com/en-us/store/b/office*

[2] Adobe Creative Cloud, individuals pay US-$ 49.99 per month. *https://www.adobe.com/creativecloud.html*

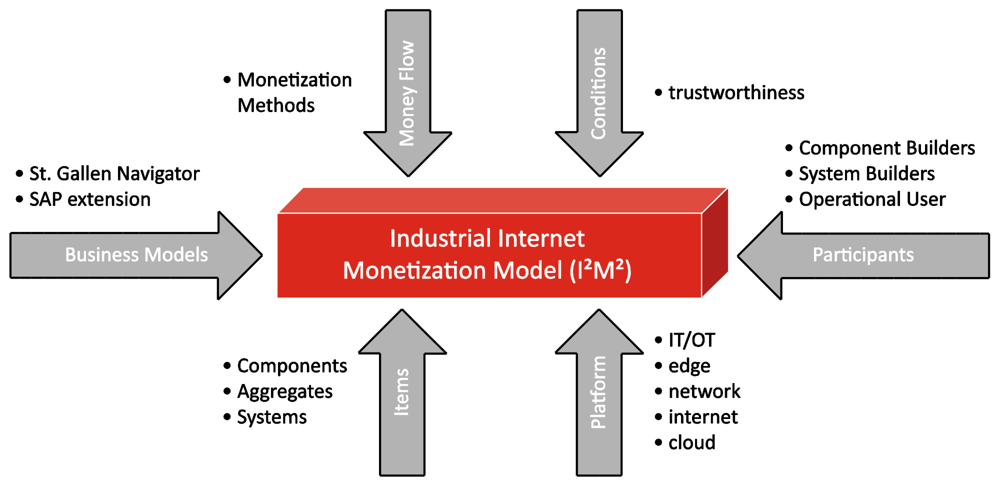

Figure 3: Parameters of I²M²

software. But the efficient pay-for-use model requires a standardized process for all components and the ability to run automatically, as illustrated as *Auto Connect* in Figure 1.

Such a standard must be open and universal. There should be no patents or other intellectual property involved. Everyone can participate and operate as shown in Figure 1. However, the industry will only accept a standard if the latter is shown to be flexible enough to fulfill the business and technical requirements of future systems for all of the participants — on the operational and builders side.

A Universal Monetization Model

Wibu-Systems discussed this idea of an open usage payment standard last year with other companies involved with the Industrial Internet Consortium (IIC). The idea was well received, but the group quickly viewed the monetization aspect in a much broader spectrum, as there are many more monetization methods possible than this one example of payment for usage.

To describe such methods, the Industrial Internet Monetization Model (I²M²) was created. Figure 3 illustrates all of the parameters of this model which are explained in detail in the next sections.

Business Models

For any specific Monetization Method, it is important to know which business models can be used: On the one hand, there are "general purpose" Monetization Methods which should be adaptable to as many business models as possible. On the other hand, there are "specialized" Methods which are intended to support very specific business methods. The selection process across business models starts with a challenge: Which business models actually exists? Luckily there is research available; one very useful list of business models is the St. Gallen Business Model Navigator

(BMILAB)[3], published by the BMILAB at the University of St. Gallen, Switzerland. The Business Model Navigator describes 55 business models by index and name — the oldest is from 1860 and the most recent from 2003. Here are a few examples:

- Franchising (#17) invented in 1860 by Singer Sewing Machine and famously copied by McDonalds and Subway.
- Razor and Blade (#39) invented in 1880 by Standard Oil Company (giving away oil lamps for free in China) but receiving its name from the Gillette business model.
- Subscription (#48) used since 1999 by companies such as Salesforce® or Netflix® to reach a steady income stream with recurring fees, typically on a monthly or annual basis.

New business models will continue to be invented and many will depend on technology which is not widely available today. SAP® for example expanded the BMILAB list by seven additional models which are specific around IoT[4]. Examples are:

- Sensor as a Service (#57) which describes revenue generated from an IoT sensor selling the measured information to other parties.
- Digitally Charged Products: Object Self Service (#56e) used by components to order independently on the internet, e.g. a heating system automatically reorders oil when necessary.

The list of business models at I^2M^2 is not limited to the lists of BMILAB and SAP, but they represent a good start.

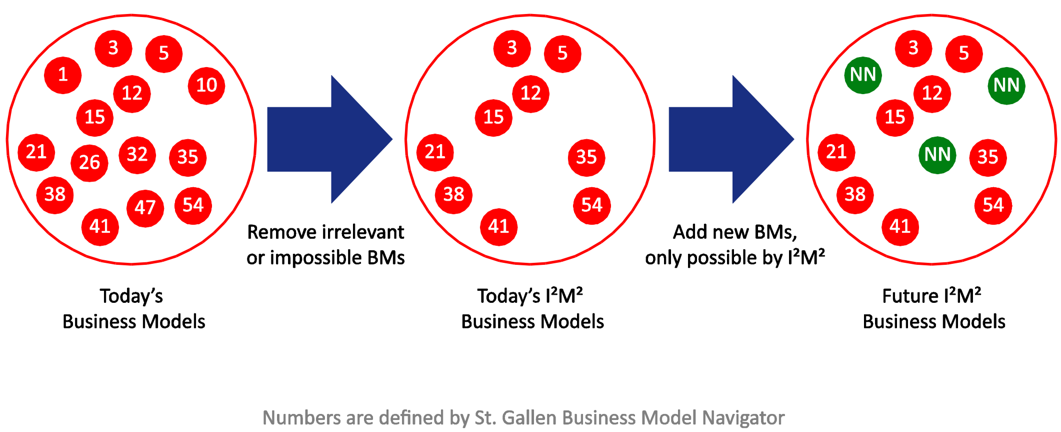

Figure 4: Selection of a group of business models inside the I^2M^2 for a specific Monetization Method

---

[3] St. Gallen Business Model Navigator, BMILAB, a spin-off from the University of St. Gallen *http://www.bmilab.com*

[4] Sung Kwon Kang, SAP Business Transformation Services: Beyond Process Innovation – BMI (Business Model Innovation), SAP Forum July 2016, Seoul, Korea *http://www.sapforum.co.kr/2016/seoul/edm/download/Track2/01.%20Business%20Model%20Innovation.pdf*

Not all Monetization Methods can be assigned to all business models; in practice, groups of possible business models are selected for specific Monetization Methods and expanded by new business models which are only possible with this method as shown in Figure 4.

A Monetization Method is *general* if it supports many business models and *special* if it is designed around one or just a few models, most likely invented specifically for this Method.

it. In Figure 5, two dark-green arrows show traditional payment paths: The Operational User pays the System Builder upfront one time and the latter pays the Component Builders one time.

Figure 5 is an abstract model of the actors. In reality, the relationship is much more complex for a specific industrial system with additional actors, shown in Figure 6. Only Monetization Methods which can be adapted to such complex relationships will have a chance for industry acceptance.

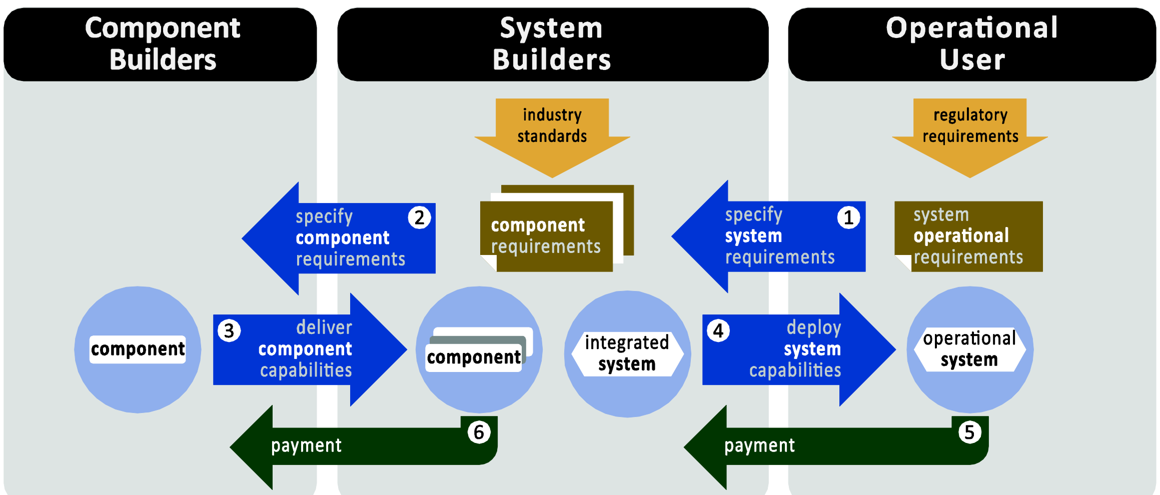

Figure 5: Schematic Actors Relationship in I²M² (adapted from (IIC-IISF))[1]

Participants

Participants are the different actors involved in an industrial system. Figure 5 explains their relationship: An Operational User wants to own or lease a new industrial system (e.g. an oil rig), creating revenue while the system is in operation. The Operational Requirements are converted by the System Builder into a real system compiling many components together into a specific design. The system is finally deployed and the Operational User operates

Monetized Items

Figure 3 shows another parameter of the I²M² model — the items which are monetized: One item is the **System** itself which is designed to use many components in a sophisticated way. In an IIoT system, all these components are ideally connected to each other and have access to the Internet.

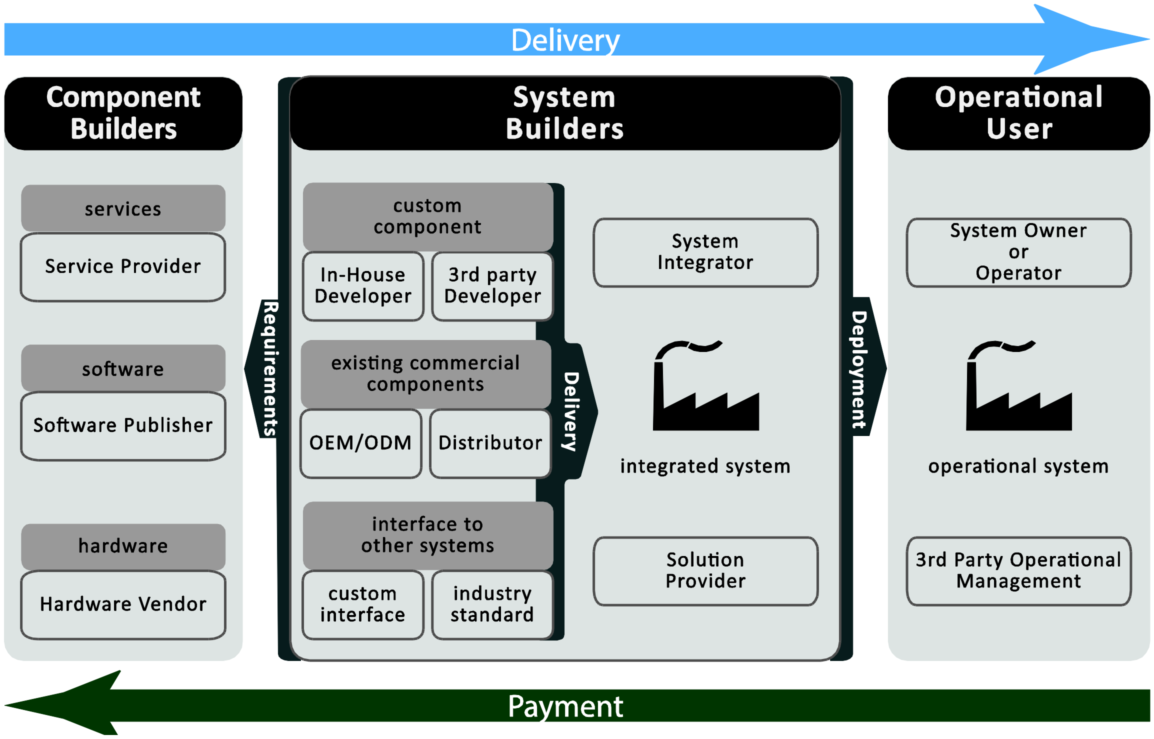

Figure 6: Detailed IIoT System Business Structure (adapted from (IIC-IISF))

As shown in Figure 6, **Components** can be hardware, software or services that, for example, are running in the cloud. These Components are not just installed in the System, side by side, but may be embedded (or stacked) into each other. Think about a Component *desktop computer* which has a hardware component *CPU board* with another component *CPU* but also has software as firmware (BIOS) and probably access to a cloud service to update the BIOS, all typically delivered from different builders. Figure 7 shows the relationship between the three types of Components and the delivery and payment flow.

The owner of this exemplary desktop computer does not see the complex money flow between the builders of all these Components but sees the paid amount for the computer when it is finally spread out during manufacturing between all these Component Builders. Traditional payment methods do not guarantee that for example the software builder of the BIOS will receive the agreed amount for each sold desktop computer, as it really depends on the honesty of the computer desktop component builder.

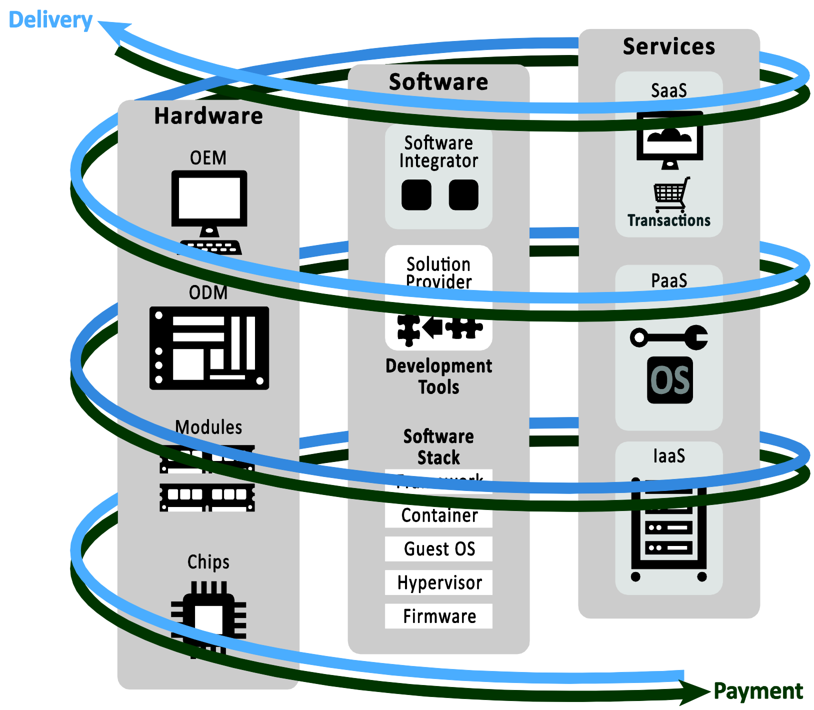

Figure 7: Schematic relationship between different components and their payment (adapted from (IIC-IISF))[1]

The I²M² defines Components containing other Components as **Aggregates.** Aggregates can be contained again in other Aggregates. For example, the described desktop computer is an Aggregate built with Components such as a power supply, hard disk and keyboard while the PC board is another Aggregate with components such as a CPU, memory and BIOS software. And this desktop computer could be part of an operational control panel, yet another aggregate.

Platforms

An IIoT system as shown in Figures 5 and 6 can be spread across several platforms, including cloud, WAN and LAN connections and areas around the edge, where the physical components are installed and operated. A major difference between such platforms is the connectivity quality, as

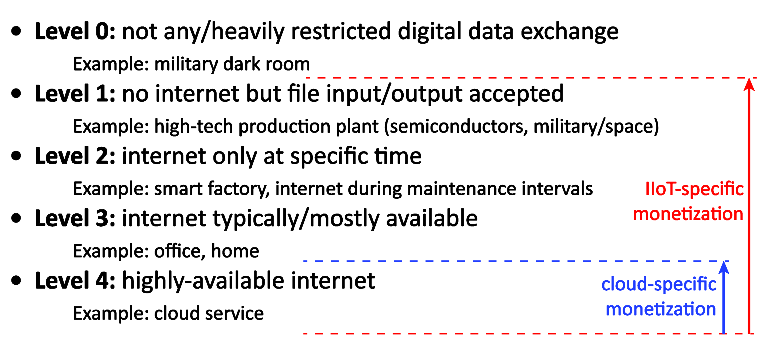

Figure 8: IIoT Connectivity Quality Model

shown in Figure 8: In the cloud platform for example, the Internet should always be available, otherwise the cloud applications cannot be operated. This high-availability (level 4) is also typical between cloud services and the WAN connections which are used to connect to the LAN and the edge. But the connectivity level from the WAN and LAN at the edge itself to the internet can be much lower:

- A typical internet connection from an office or a plant is not necessarily redundant, so the internet is typically available, but in many cases with no guaranteed minimum down time, defined as Level 3 of the Connectivity Quality.
- Many Operational Users of IIoT systems refuse typical permanent internet connectivity from a production or a power plant, for example, due to security reasons. In this case, they may want to restrict the internet access only during maintenance intervals, for example. Such "limited by demand" availability is defined as Level 2.
- Even more careful Operational Users additionally restrict the format which can be used for data exchanged across the internet. One example is only permitting the exchange of files via special high-security gateways. Our model defines this as connectivity quality level 1.
- The most restricted connectivity—no access at all—is defined as level 0 and is not considered in the I²M² because an IIoT operation cannot be established in this case.

But it should be clear that any widely accepted industrial Monetization Method must support more than just Level 4. So, typical payment methods which are established in the cloud and also frequently used with connected cell phones cannot be automatically used in the IIoT systems.

Conditions

An industrial system operates in the real world and its specific demands influence any Monetization Methods as well. Most operational conditions dictate the level of trustworthiness of this system [5] **Error! Reference source not found.**. For example, safety regulations as part of trustworthiness could demand that a safety-related component must also work even if its subscription was not payed.

Monetization Methods

The most important parameter of the I²M² is the Monetization Method itself. Ideally the Model can describe many of such methods and the list of methods is open-ended for the future. But all of these methods use the other parameters of I²M², so they can also be compared to each other relatively easily. For example, the minimum IIoT Connectivity Quality Model Level (Figure 8) could be a fixed parameter for a specific method.

[5] IIC: The Industrial Internet, Volume G4: Security Framework Technical Report, version 1.0, 2016-Sep-26, retrieved 2017-01-10 *http://www.iiconsortium.org/IISF*

The following list gives a short introduction to several Monetization Methods; one of these will be the central focus of the subsequent sections of this article:

- **Upfront Monetization:** This is the traditional monetization method and easily explained: The System Builder (Figure 5) buys all the components and pays the component manufacturers at time of delivery, at the latest. For the design and installation of the system, a surcharge is added to the sum of payments of all components; the result is the final system price which is paid by the Operational User by the time the system is ready to operate. Banks are typically involved in the process to provide short-term and long-term loans.
- **Dynamic Monetization:** This is a new model, today only used in the cloud or by some IT computer software. In principle, the Operational User pays the System and Component Builders partially upfront and the balance *during* the operation of the system. This Method was introduced at the beginning of this article and will be explained in detail in the subsequent sections.
- **Independent Analytic Monetization**: The System Builder or a third-party company collects operational data from the running IIoT system, analyzes it and sells it as anonymous analytic data to competitors of the System Builder who receives a share of this revenue. Such data can be used to optimize the efficiency of comparable industrial systems (benchmarking) in exchange for lower operation cost for the User by providing the data.
- **Data Aggregation Monetization**: Similar to Independent Analytic Monetization, but the analytic does not start until the system reaches the sunset at the end of life. Such data can be used to optimize the redesign of future similar systems of the System User and competitors; the User again gets a share of the revenue.

Having several methods available during the design of a specific IIoT system provides great flexibility for the participants and actors shown in Figure 5: Most likely in the future, a monetization or payment plan for a specific IIoT System will consist of a combination of several Monetization Methods.

DYNAMIC MONETIZATION METHOD

The rest of this article takes a deeper look at the structure, opportunities and challenges of the Dynamic Monetization Method. At the beginning of this article the overall business advantages of this Monetization Model were explained. Now let's see how such a method could be practically implemented.

First, the overall costs of an industrial system are broken down into four cost classes:

- **Physical Costs**: Required to build the component from raw material or other components.
- **Intellectual Costs**: Required to design a component or a system in addition to the Physical Costs.

- **Setup Costs**: Required to install, test and validate the system.
- **Profit**: The component and system builders' reward for business success.

To explain further, this table provides some examples of the Physical and Intellectual Costs of IIoT device components:

Cost Type	Examples for IIoT devices
Physical	electronic boards, wiring and connectors, computer chips, basic displays, power supplies
Intellectual	Developed software, software libraries, software platforms (OS, connectivity), patents, cloud services, intellectual properties such as pictures, sounds, videos etc., share of research and crosscutting design and development

Physical Cost can also be understood as the minimum payment to avoid a direct "loss by delivery" for the component builders. The traditional industry uses such cost in insurance cases: It is quite similar to the "cost of replacing a component". If for example your computer was lost in a fire, the insurance company replaces the hardware but not the installed data or software, which can be recreated from backup or license keys, both secured separately.

The changing ratio between the hardware and software cost of a device from Figure 2 can be transferred directly to a similar ratio between physical and intellectual costs: For smart devices in particular, the intellectual cost will rise while the physical cost will stay flat or even fall.

UPFRONT AND USAGE COSTS

The Dynamic Monetization Method defines Physical and Setup Costs as **Upfront Costs** and Intellectual Costs and Profit as **Usage Costs**. Upfront Costs are traditionally paid before the system is deployed while the Usage Costs are paid when the system is operated and creates revenue for the Operational User. Upfront Costs principally

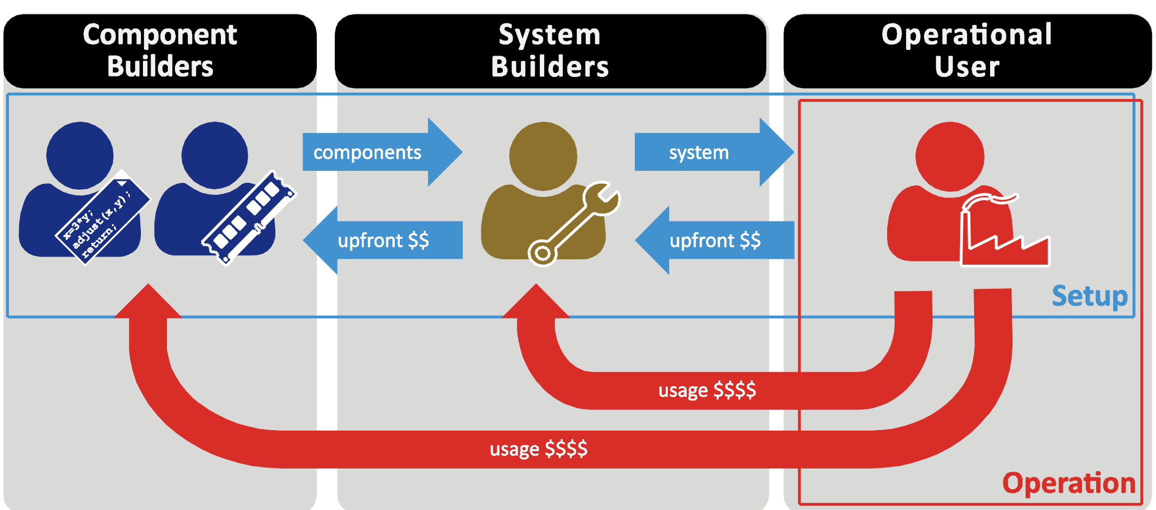

Figure 9: Split payment between upfront and usage cost

follow the Upfront Monetization Method, explained in the section above. Figure 9 illustrates the physical delivery and setup with the upfront payment highlighted with blue arrows and the usage payment with red arrows. The latter is directly transferred from the Operational User to the Component Builders, not via the System Builder.

Advantages and Challenges of the Dynamic Monetization Method

The biggest advantage for the Operational User is the significant reduction of the upfront costs traditionally paid before collecting the revenue generated from operating the system. The usage payment is quite stable over time, so ROI can easily be predicted and calculated.

A challenge for the Operational User is that he needs to know the potential Usage Costs before he orders the system. This requires a comprehensive Usage Cost simulation of all components involved. The Dynamic Monetization Method needs to provide a mechanism to make such simulations as simple and automatic as possible. Another challenge is that any Component Builders could simply increase the Usage Costs after the system is delivered — keeping the Operational User deeply dependent upon these suppliers. As a result, the Dynamic Monetization Method must provide a "smart price table" system for all installed components, which not only talks about the current Usage Costs but also provides a future price guarantee or a maximum range of predictable price increase.

A challenge for the Component and System Builders is that they do not get paid instantly for all of their research and development efforts, which are used across components and the system. And their profit is part of the Usage Cost too. If the Operational User decides to shut down the system too early, they will never receive the income stream from the Usage Cost. On the other hand, there are no more "discounted" systems, such as in cases where the original upfront costs were too high and builders desperately try to close the deal by offering to lower the system cost. And if the system operates as expected, all builders will receive an ongoing usage revenue stream, which over time will finally exceed the original payment of full-upfront costs. In general, it provides a closer bond between Builders and Operational User in the sharing of risks and opportunities.

Usage payments reduce the amount of required loans and eliminate the additional interest payments. In a government tax model, usage payments will be frequently paid as a cost of operation (comparable with that for power and gas) instead of an investment which needs a depreciation calculation.

There are also advantages for human society and the environment: Old systems are more rapidly replaced with modern and probably more expensive components because the upfront costs will be lower. Savings in energy or the reduction of payments for pollution penalties can easily pay for the Usage Costs of such components.

Expensive high-tech medical devices, such as computed tomography (CT) scanners, are a good example of how the Dynamic Monetization may provide advantages for consumers and society:

- Today these machines are typically sold with a 100% Upfront Cost. The high payments force operators to use these machines as much as possible: CT scans are expensive, and if performed unnecessarily, contribute to the rising costs in our health system.
- If machines like this could be delivered with minimal Upfront Cost and the Usage Cost paid as a share of each scan, the operators could scale back the use of the scanner to only necessary cases.
- The builder of the CT scanner would likely earn less money per installation, but due to lower Upfront Cost, more hospitals could afford newer scanners, resulting in the same or even higher income for the supplier.
- If CT scanners are more affordable due to lower Upfront Cost, smaller hospitals in rural areas would then have easier access to modern scanners, providing better health care and possibly saving lives.

In summary the Dynamic Monetization Method has some challenges but the advantages will exceed the risks. And the software industry has already demonstrated feasibility of this approach.

Dynamic Monetization Challenges

For the Dynamic Monetization Method itself there are also many business and technical challenges which require sophisticated solutions. Two of these challenges are described below.

Dynamic Monetization of Aggregates

Aggregates, defined as components that include sub-components from different builders, are a challenge for the Usage Cost payments: Every Component Builder would receive the individual Usage Cost from the Operational User, even if the component is deeply hidden in the aggregate. But the latter wants to know before he orders such an aggregate what the sum of Usage Costs for all the Components of this Aggregate will

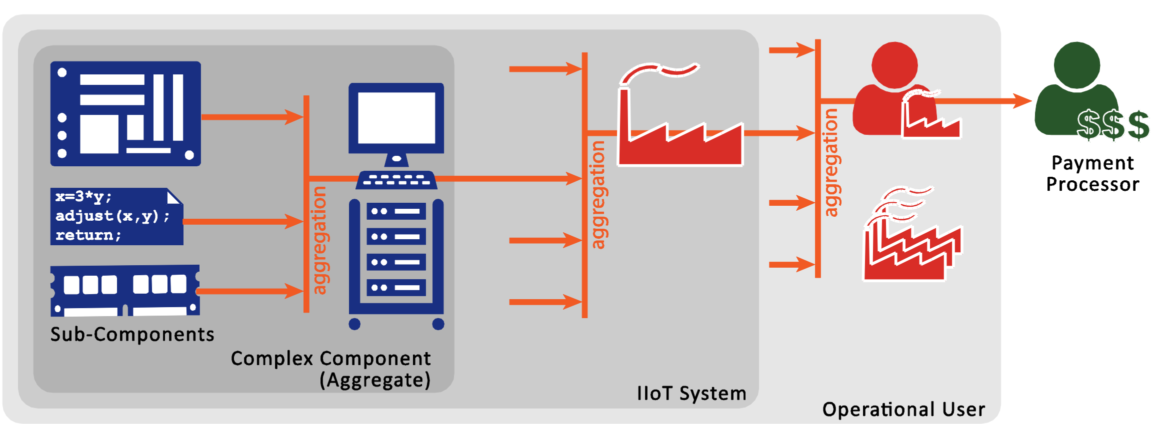

Figure 10: Dynamic payment of aggregates

be. In reality, the builder of an included component negotiates a fixed price with the aggregate builder for the Usage Costs. Depending upon the negotiation, for the same Component, this price can be different for different Aggregates. If this Component now connects to the automatic payment process (see Figure 1), this connection needs to be routed through the aggregate, so the correct price of this Component in this Aggregate can be defined, as shown in Figure 10. This figure also shows that this price adjustment can be expanded to the whole System, making it possible for a large Operational User to have (previously negotiated) different Usage Costs for principally the same system at different locations, independent of areas, countries, etc. In this situation, each System aggregates all Usage Cost requests from all components and makes specific, previously negotiated price corrections before the System applies for the total Usage Costs.

Dynamic Monetization with Low IIoT Connectivity Quality

Another challenge for the Dynamic Monetization Method is the Connection Quality, explained earlier and in Figure 8: It is not realistic to expect that an industrial system is always online and able to send payment information to the Payment Processor in the cloud during usage (see Figure 1). Having a method such as "you pay while you use the component and we track usage online" as shown in the upper half of Figure 11 is *not* realistic. A token-based solution, as illustrated in the lower part of Figure 11, is more promising: If the System is online, the payment is established and the Payment Processor returns tokens for each paid component, enabling it to run it offline for a specific interval, e.g. a month or even a year. If the token expires, the component cannot be used any longer, so the token has to be renewed frequently while the system is connected to the Internet. With some technical methods, it is even possible to

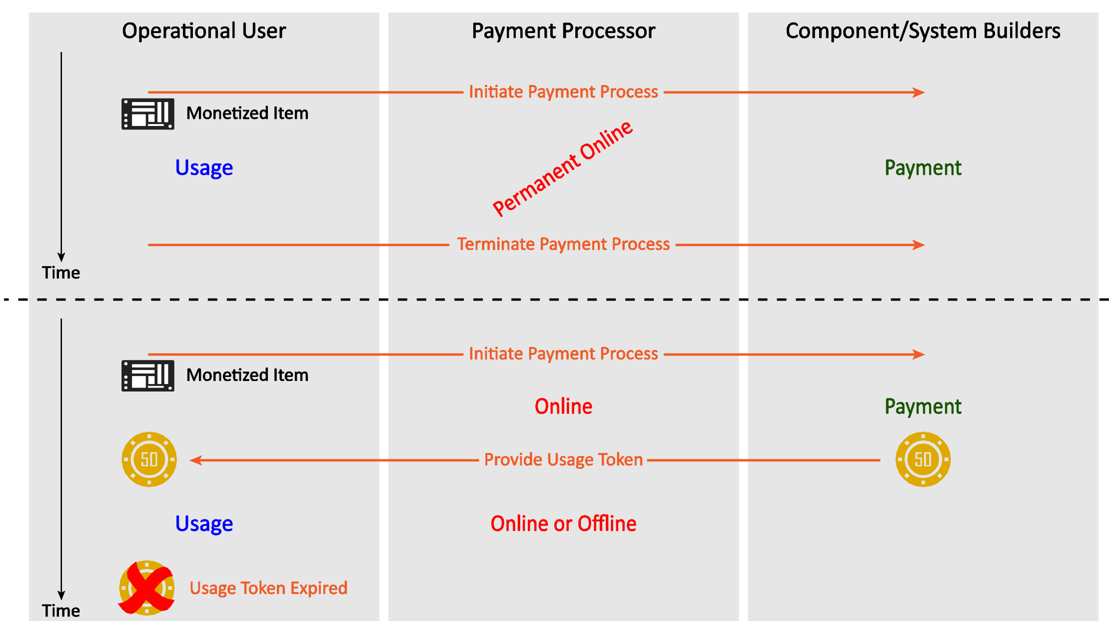

Figure 11: Status management of dynamic monetization method

implement the whole payment/token process for a complete system with two (probably large) files, sent and received between system and payment processor, so even the Connectivity Quality Level 1 of Figure 8 can be realized.

Summary

This article is only an introduction to the complex but also exciting world of Industrial Internet Monetization. Using I^2M^2 as a model for different Monetization Methods helps to establish such methods in the future. The different scenarios can illustrate how they might fit within a world of varying business models, components and systems and the relationships with different participants such as Operational Users, Component and System Builders.

One of the promising Monetization Methods is Dynamic Monetization. Most significantly, it moves the payment for industrial systems from upfront to the time of usage. This method faces challenges from markets and participants, but all can be solved with a sophisticated standard. The advantages of Dynamic Monetization will redefine how industrial systems will be paid in the future.

Future Steps

At the IIC, only the Business Viewpoint of I^2M^2 has been described to date. A planned whitepaper will describe the Model and several Methods with all the business aspects (see Figure 12).

After all of the business requirements have been discussed, the IIC will move on to an Implementation Viewpoint, which describes all communication standards necessary to implement the model shown in Figure 1. Once all of this information is collected, it will be possible to define a plug-and-play standard, which will finally automate the Dynamic Monetization method as much as possible, so that the vision can become reality.

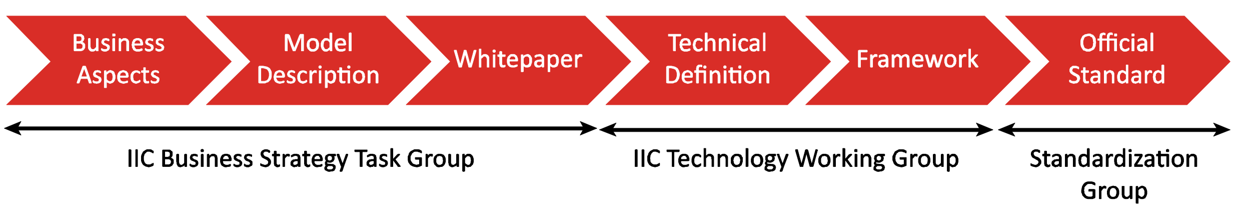

Figure 12: planned future progress of I^2M^2

➢ Return to IIC Journal of Innovation landing page for more articles and past editions.